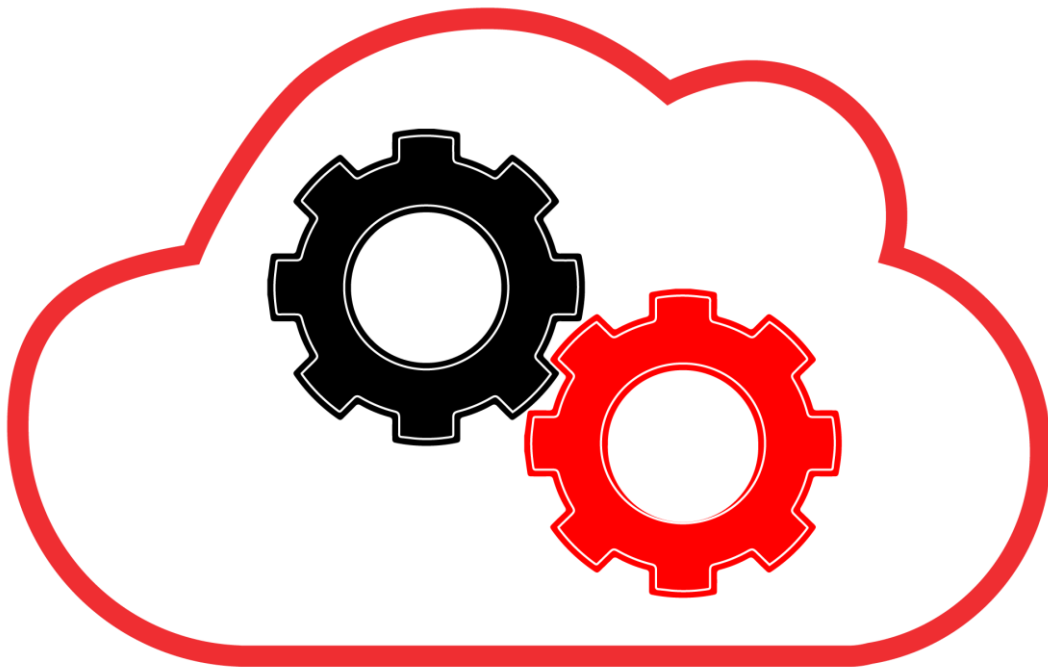


OPEN SHIFT



Primeros Pasos

Eirc Gustavo Coronel Castillo
gcoronelc@gmail.com



INDICE

CAPÍTULO 1 CREACIÓN DE UNA CUENTA	4
SITIO WEB	4
CREACIÓN DE CUENTA	5
INICIO DE SESIÓN	6
CAPÍTULO 2 INSTALACIÓN DE GIT	8
OBTENER EL SOFTWARE	8
PROCESO DE INSTALACIÓN	8
COMANDOS INICIALES	13
<i>Verificar la instalación de Git</i>	<i>13</i>
<i>Consultar la versión de Git instalada</i>	<i>13</i>
<i>Registra tu nombre</i>	<i>14</i>
<i>Registra tu correo electrónico</i>	<i>14</i>
CREAR UN REPOSITORIO EN GITHUB	15
<i>Crear una cuenta en GitHub</i>	<i>15</i>
<i>Procede a crear un repositorio</i>	<i>16</i>
CLONAR REPOSITORIO DEMO	17
EDITAR REPOSITORIO	19
<i>Modificar el archivo .gitignore</i>	<i>19</i>
<i>Crea un archivo</i>	<i>19</i>
ACTUALIZAR EL REPOSITORIO EN GITHUB	20
<i>Verificar el estado del repositorio</i>	<i>20</i>
<i>Confirmar los cambios en el repositorio local</i>	<i>20</i>
<i>Subir los cambios a GitHub</i>	<i>21</i>
ACTUALIZAR EL REPOSITORIO LOCAL	23
<i>Modifica el archivo into.txt en GitHub</i>	<i>23</i>
<i>Actualiza tu repositorio local</i>	<i>24</i>
CAPÍTULO 3 INSTALACIÓN DE RUBY	25
OBTENER SOFTWARE	25
INSTALACIÓN	26
CAPÍTULO 4 RHC CLIENT TOOLS	28
RHC	28
INSTALACIÓN	28
CONFIGURACIÓN DE RHC	29
<i>Comando a Utilizar</i>	<i>29</i>



<i>Paso 1: Ejecutar Comando</i>	<i>29</i>
<i>Paso 2: Hostname del Servidor.....</i>	<i>29</i>
<i>Paso 3: Inicio de Sesión</i>	<i>29</i>
<i>Paso 3: Generación de Token</i>	<i>30</i>
<i>Paso 4: Generación de Llaves SSH</i>	<i>30</i>
<i>Paso 4: Recomendaciones Finales</i>	<i>31</i>
 CAPÍTULO 5 CREACIÓN DE UNA APLICACIÓN	 33
CREACIÓN DE LA APLICACIÓN	33
PUBLICAR UN WAR.....	35

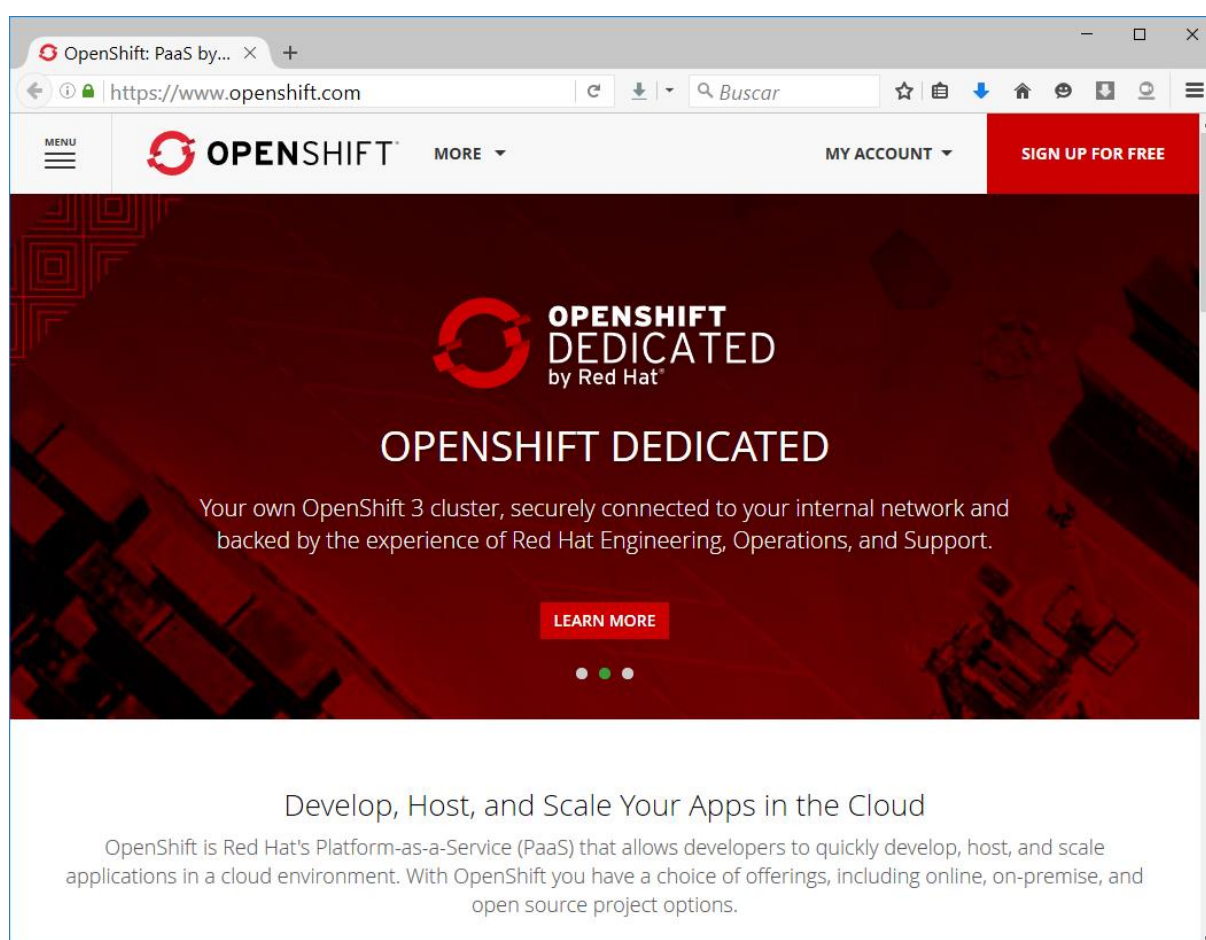


Capítulo 1

CREACIÓN DE UNA CUENTA

SITIO WEB

<https://www.openshift.com/>



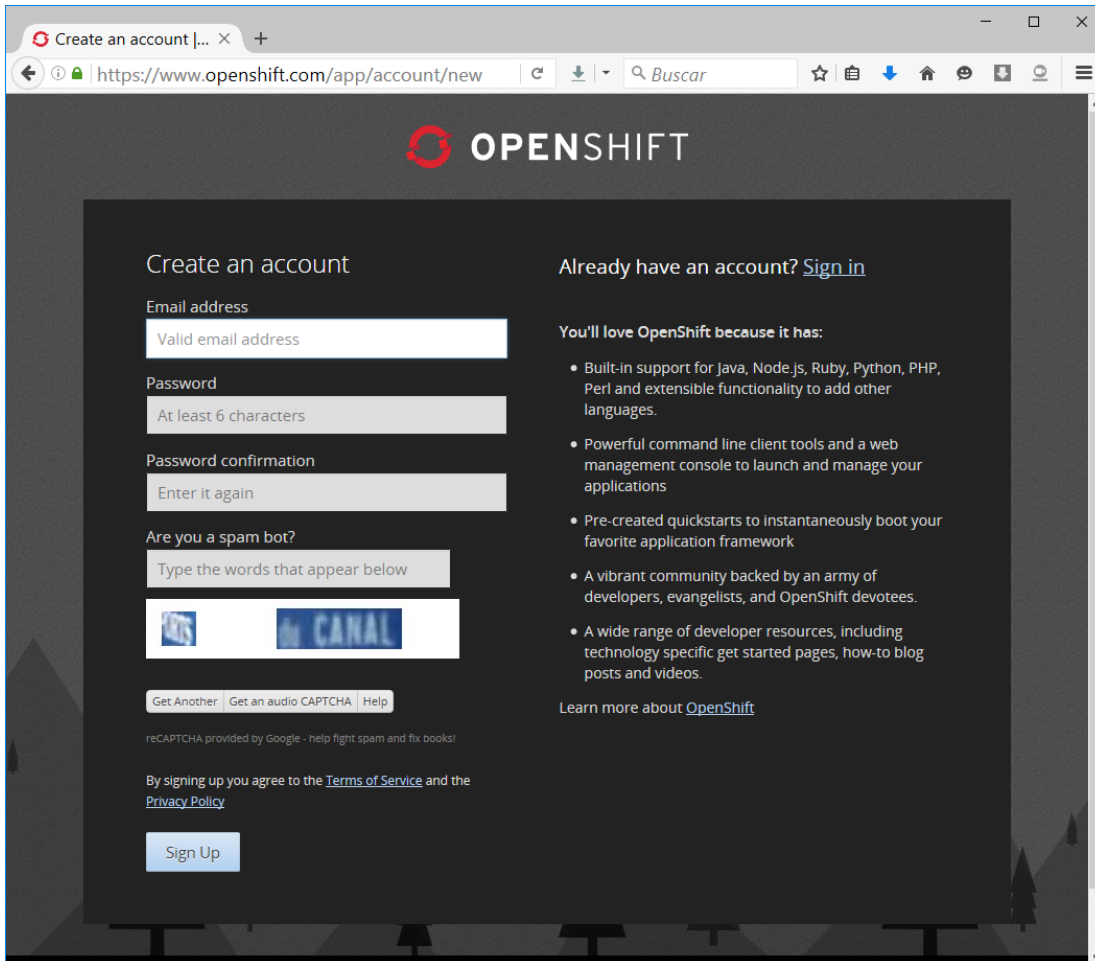


CREACIÓN DE CUENTA

Hacer click en la opción:

SIGN UP FOR FREE

Luego proceda a ingresar sus datos:



Después de ingresar los datos solicitados haga click en el botón:

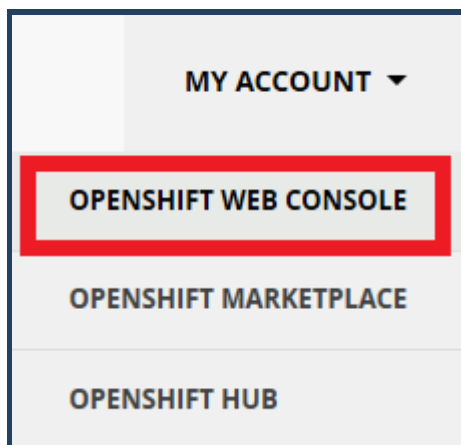
Sign Up

Después recibirás un correo para que confirmes la creación de tu cuenta, este paso es necesario para que puedas empezar a crear tus aplicaciones.

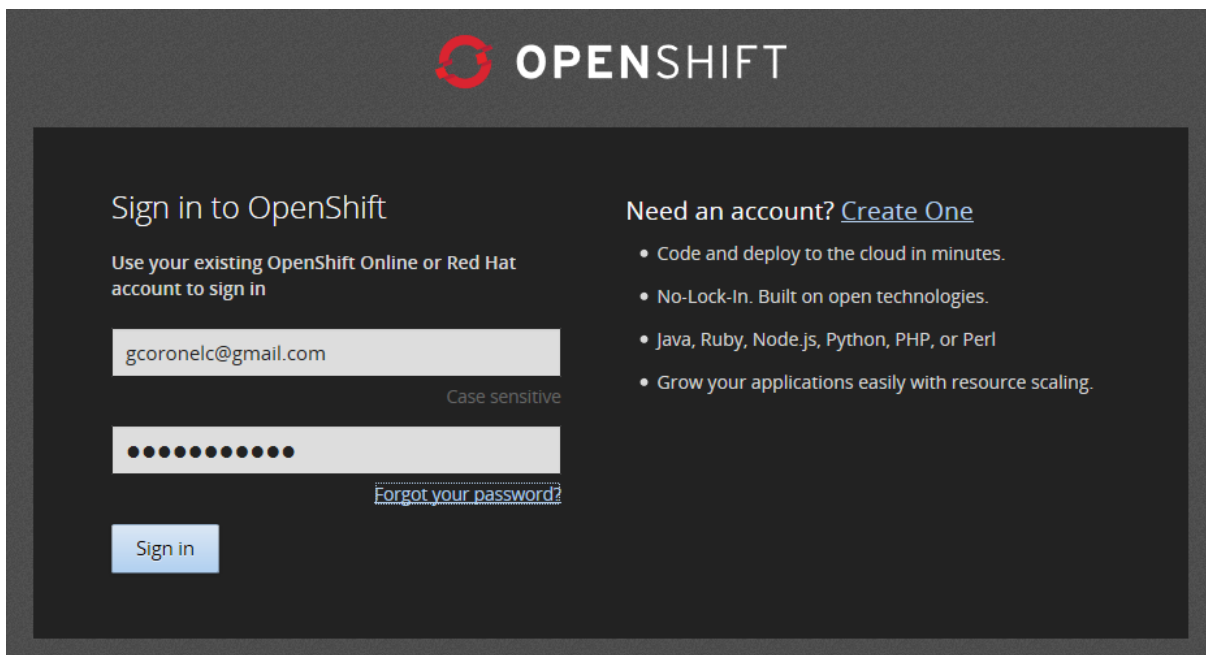



INICIO DE SESIÓN

Para iniciar sesión en el portal de **OpenShift** de ir a la opción **MY ACCOUNT** y ejecutar la opción **OPENSIFT WEB CONSOLE**.



Luego debes ingresar tu correo electrónico y tu clave para que inicies sesión.



 **OPENSIFT**

Sign in to OpenShift

Use your existing OpenShift Online or Red Hat account to sign in

Case sensitive

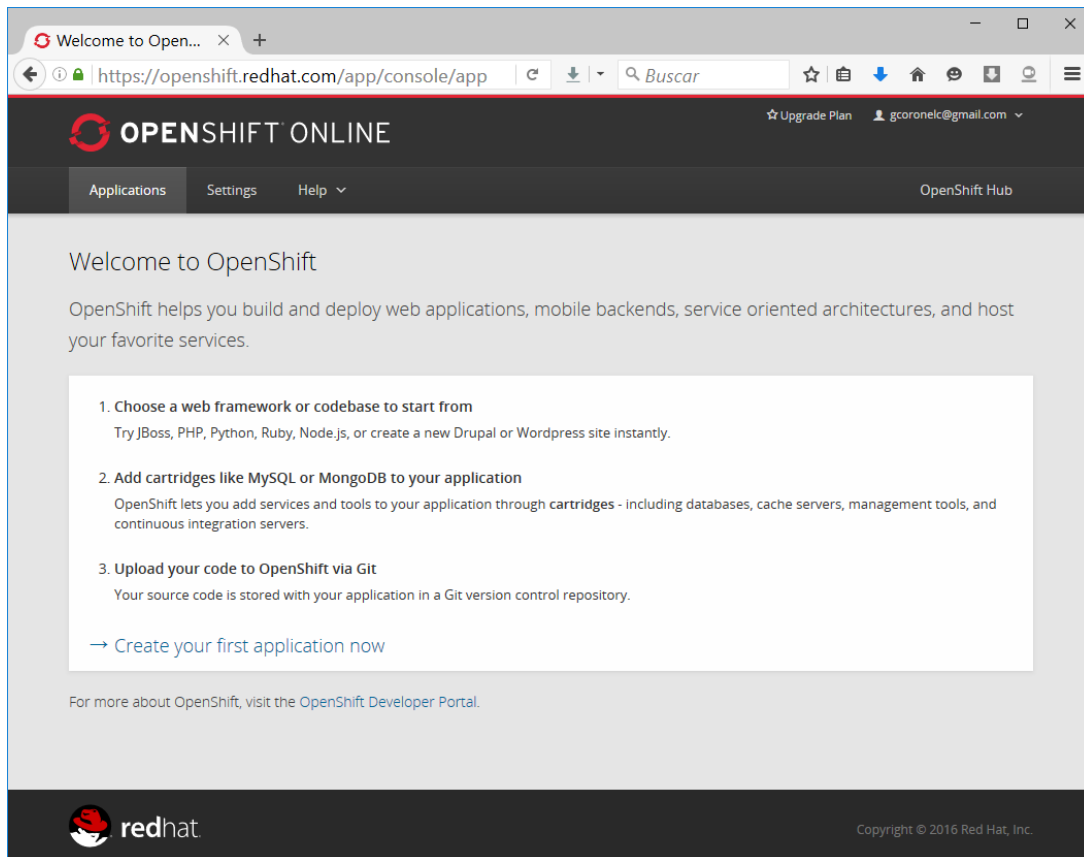
[Forgot your password?](#)

Need an account? [Create One](#)

- Code and deploy to the cloud in minutes.
- No-Lock-In. Built on open technologies.
- Java, Ruby, Node.js, Python, PHP, or Perl
- Grow your applications easily with resource scaling.



Esta es la bienvenida de la aplicación:





Capítulo 2 INSTALACIÓN DE GIT

OBTENER EL SOFTWARE

Te recomiendo que instales **GIT FOR WINDOWS**, la ruta es la siguiente:

`https://git-for-windows.github.io/`

El archivo que debes descargar para Windows de 64 Bits es el siguiente:

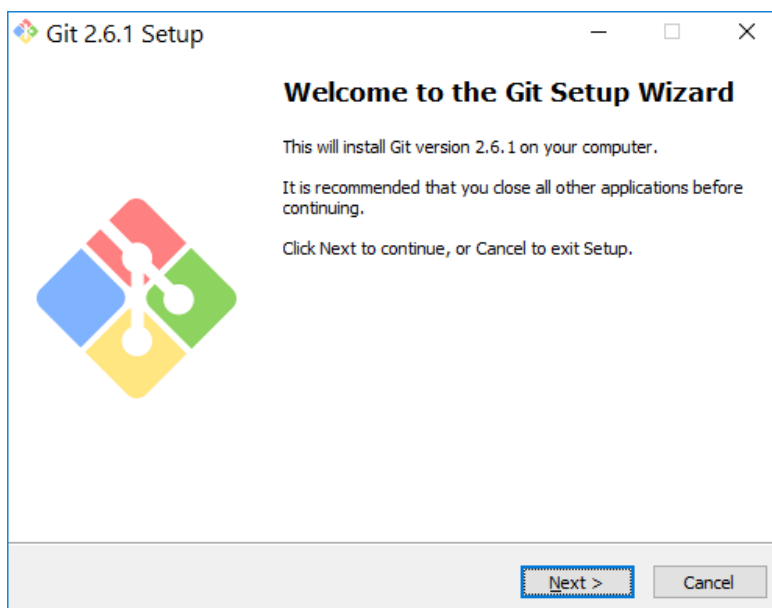
`Git-2.6.1-64-bit.exe`

Es posible que en este momento que estás leyendo este documento ya exista una nueva versión.

PROCESO DE INSTALACIÓN

Durante el proceso de instalación debes integrarlo con la consola de Windows.

1. Ventana de bienvenida, solo debes hacer click en el botón **Next**.

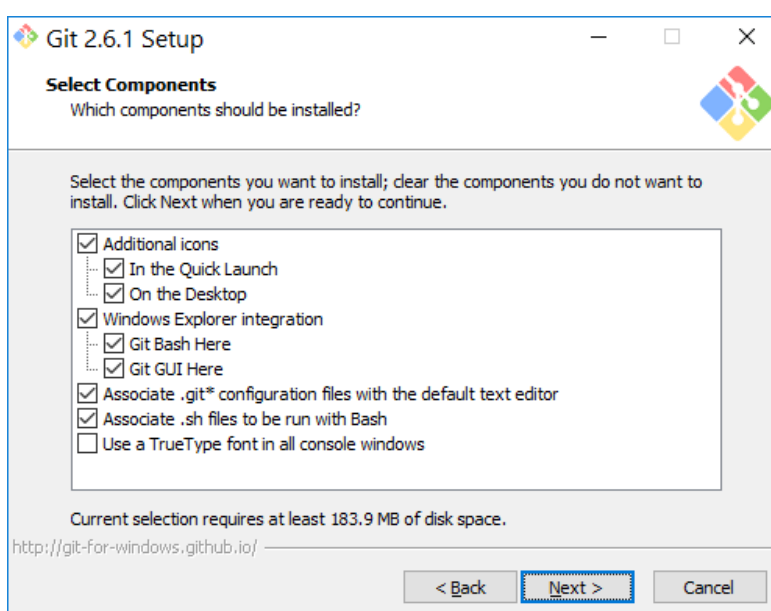




2. Ventana de licencia del software, solo debes hacer click en el botón **Next**.

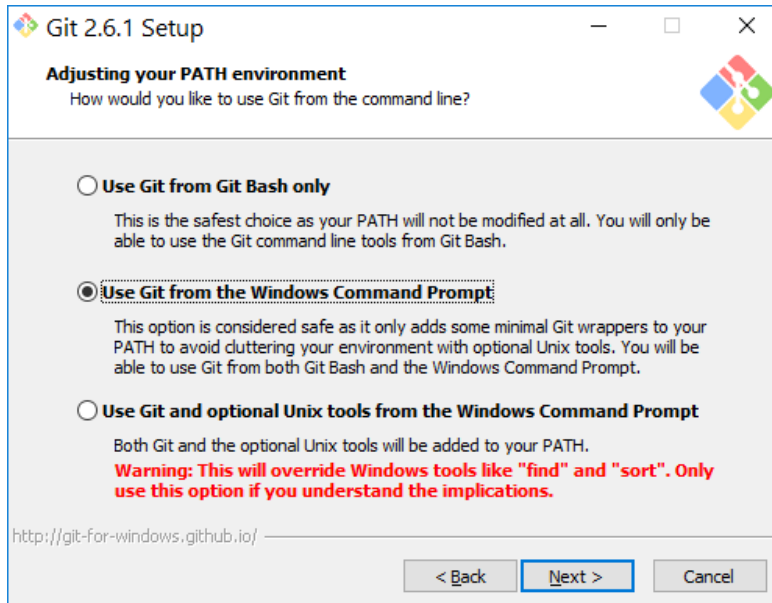


3. Ventana de componentes a instalar, seleccione los componentes tal como se ilustra en la siguiente imagen y haga click en el botón **Next**.

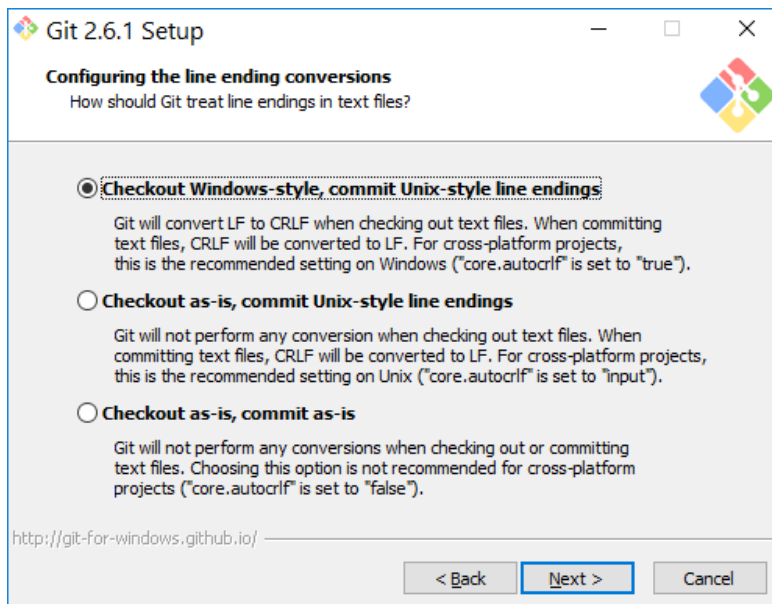




4. Ventana de configuración de la variable PATH de entorno de Windows. Seleccione la opción tal como aparece en la siguiente imagen y haga click en el botón **Next**.

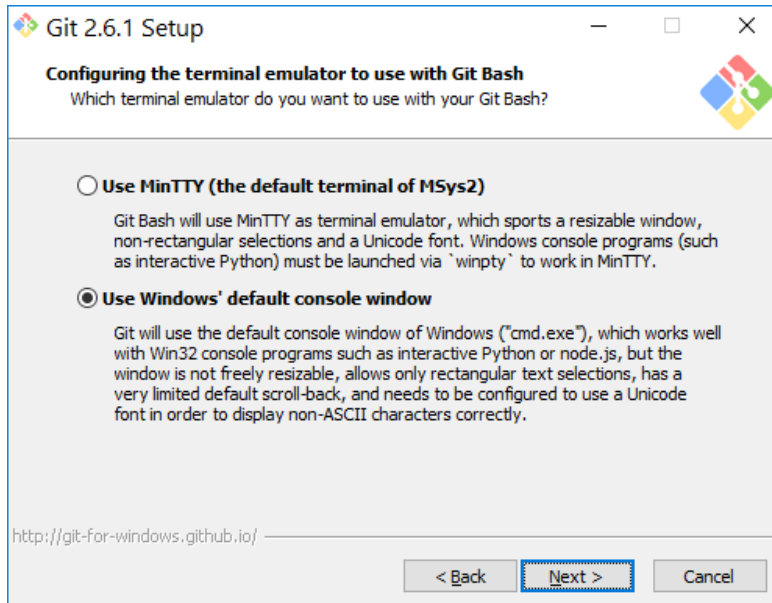


5. Ventana de configuración de conversiones de fin de línea. Deje la opción por defecto y haga click en el botón **Next**.

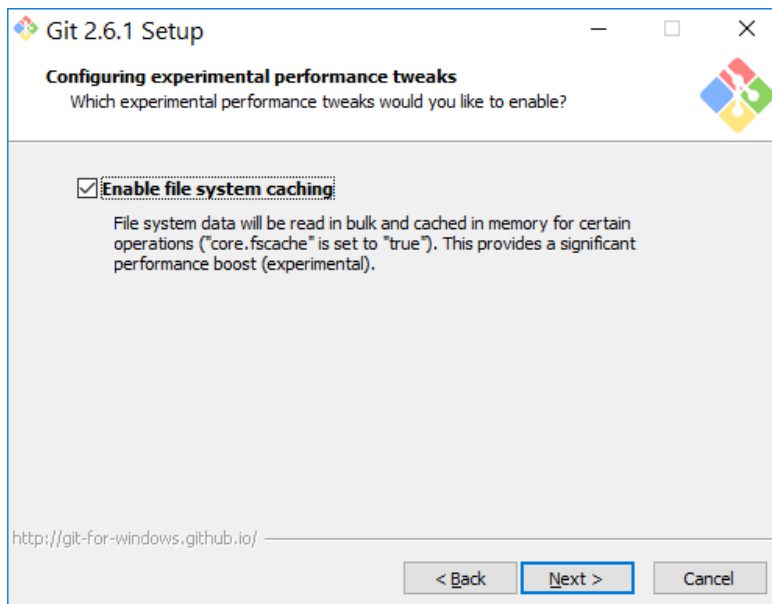




6. Ventana de configuración de terminal. Seleccione la opción que aparece en la siguiente imagen y haga click en el botón **Next**.

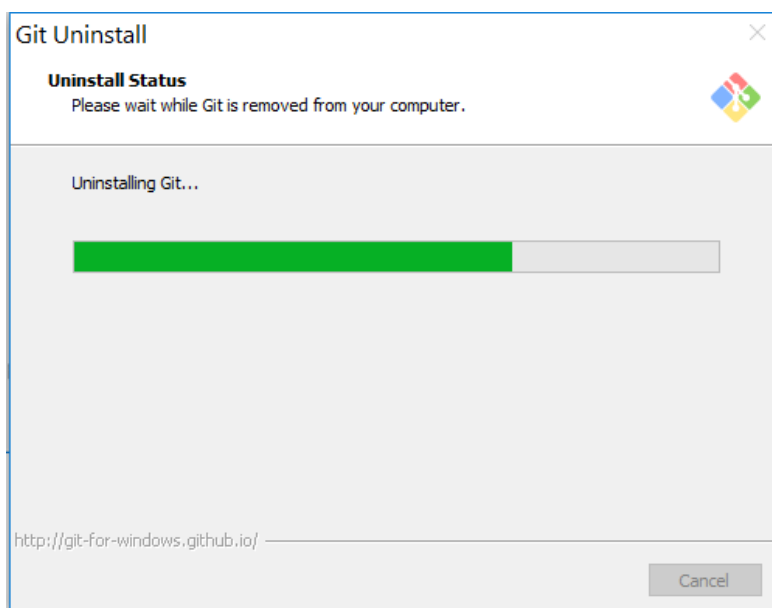


7. En la siguiente ventana configure el uso de sistema de cache y haga click en el botón **Next**.

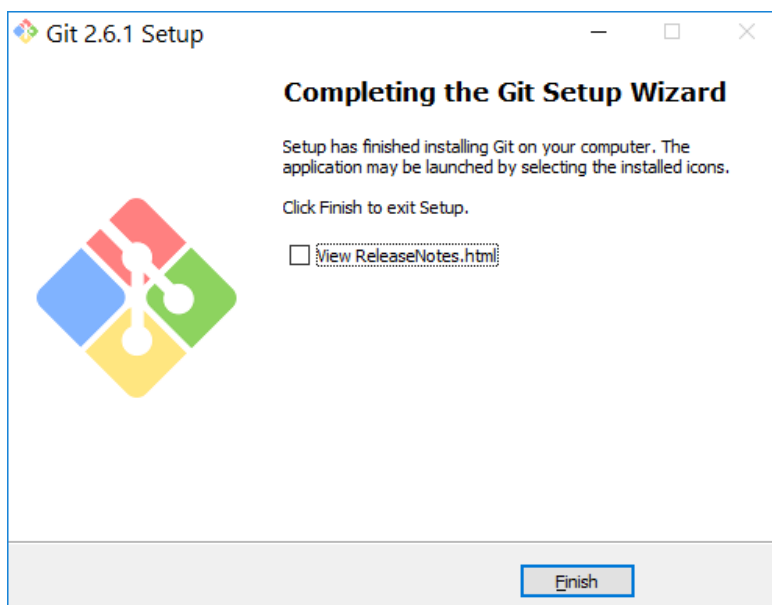




8. Luego se inicia el proceso de instalación.

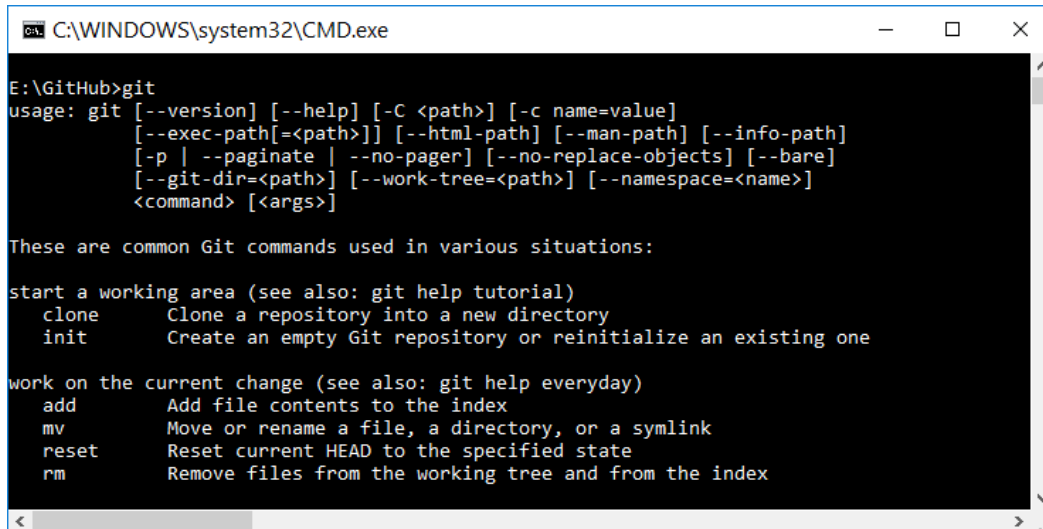


9. Finalmente, el proceso de instalación finaliza, haga click en el botón **Finish**.



COMANDOS INICIALES

Todos los comandos se deben ejecutar en la consola de Windows. Personalmente, tengo una carpeta GitHub donde tengo todos mis repositorios que mantengo en GitHub.



```
C:\WINDOWS\system32\CMD.exe

E:\GitHub>git
usage: git [--version] [--help] [-C <path>] [-c name=value]
       [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
       [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
       [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
       <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset      Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index
```

Verificar la instalación de Git

Cuando ejecutas el comando **git** muestra su sintaxis y una ayuda sobre cada uno de los comandos que puedes ejecutar para gestionar un repositorio.

```
E:\GitHub>git
usage: git [--version] [--help] [-C <path>] [-c name=value]
       [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
       [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
       [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
       <command> [<args>]
```

Consultar la versión de Git instalada

```
E:\GitHub>git --version
git version 2.6.1.windows.1
```



Registra tu nombre

Es muy importante que registres tu nombre para saber quién o quienes estan registrando cambios en el repositorio.

La sintaxis es la siguiente:

```
git config --global user.name "Aquí escribe tu nombre y apellido"
```

Por ejemplo, para mi caso sería así:

```
E:\GitHub>git config --global user.name "Eric Gustavo Coronel Castillo"
```

Registra tu correo electrónico

Es muy importante que registres tu correo electrónico para saber quién o quienes están registrando cambios en el repositorio.

La sintaxis es la siguiente:

```
git config --global user.email "Aquí escribe tu correo electrónico"
```

Por ejemplo, para mi caso sería así:

```
E:\GitHub>git config --global user.email "gcoronelc@gmail.com"
```




CREAR UN REPOSITORIO EN GITHUB


Crear una cuenta en GitHub


En GitHub (<https://github.com>) procede a crear tu cuenta de usuario.

Join GitHub

The best way to design, build, and ship software.

 **Step 1:**
Set up a personal account

 **Step 2:**
Choose your plan

 **Step 3:**
Go to your dashboard

Create your personal account

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

[Create an account](#)

You'll love GitHub

- Unlimited** collaborators
- Unlimited** public repositories
- ✓ Great communication
- ✓ Friction-less development
- ✓ Open source community

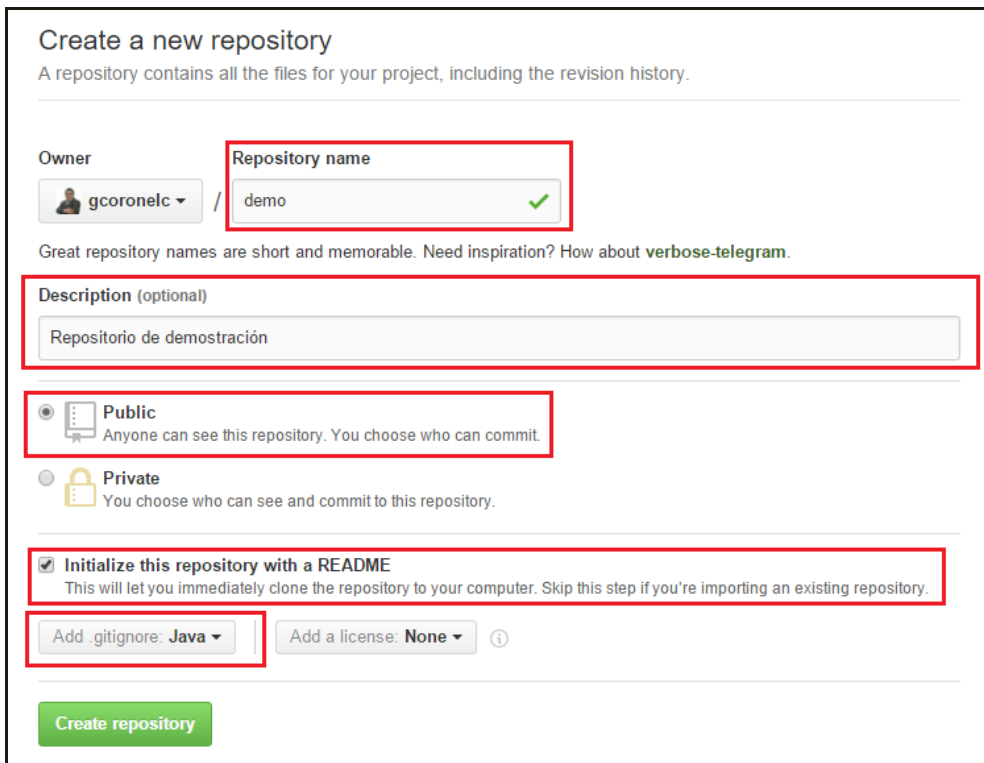
GitHub te enviará un correo electrónico para que confirmes la creación de la cuenta.



Procede a crear un repositorio

Procede a crear un repositorio de demostración de nombre **demo**.

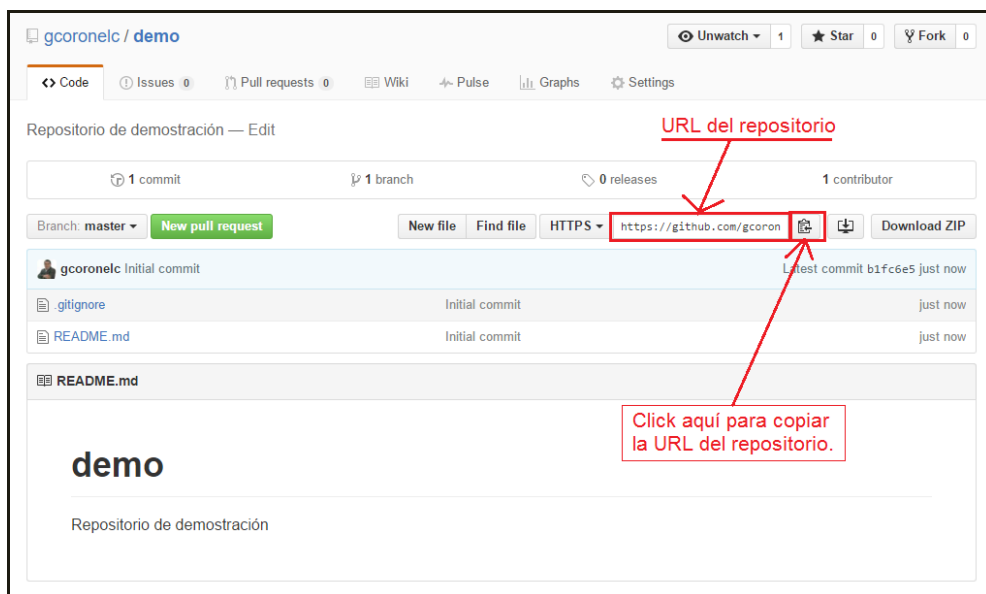
La siguiente imagen se ilustra la creación de repositorio **demo** con mi cuenta.



The screenshot shows the 'Create a new repository' form on GitHub. The form includes the following fields and options:

- Owner:** gcoronelc
- Repository name:** demo (with a green checkmark)
- Description (optional):** Repositorio de demostración
- Visibility:** Public (selected), Private
- Initialize this repository with a README:** (checked)
- Add .gitignore:** Java
- Add a license:** None
- Create repository:** (green button)

A continuación tienes la imagen que confirma la creación del repositorio **demo**.



CLONAR REPOSITORIO DEMO



Ahora vas a clonar el repositorio **demo** creado en el ítem anterior.

Para clonar un repositorio la sintaxis es:

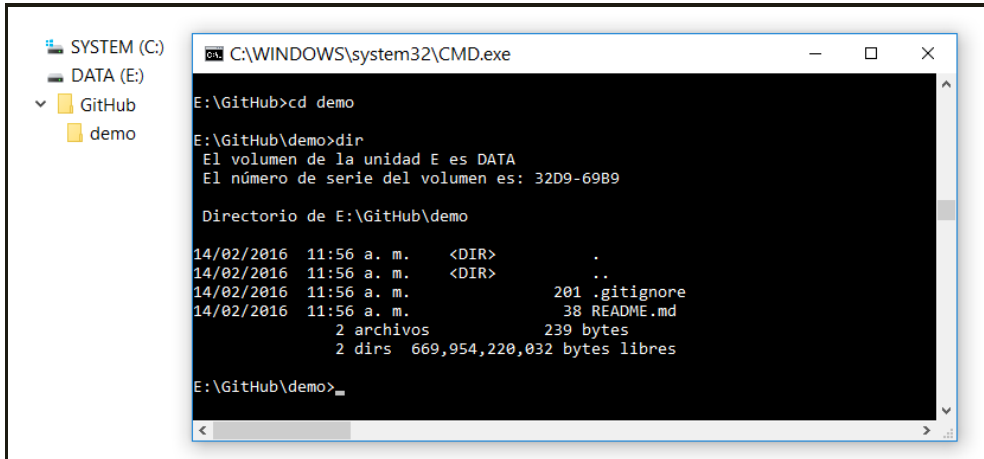
```
git clone <URL del repositorio>
```

Para el caso del repositorio **demo**, sería así:

```
E:\GitHub>git clone https://github.com/gcoronelc/demo.git
Cloning into 'demo'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
Checking connectivity... done.
```



El repositorio ha sido clonado correctamente, no presenta ningún mensaje de error. En la unidad E: puedes verificar que se ha creado la carpeta demo correspondiente a repositorio, tal como se ilustra en la siguiente imagen:



The screenshot shows a Windows File Explorer window. On the left, the 'FOLDER' pane shows the hierarchy: SYSTEM (C:) > DATA (E:) > GitHub > demo. The main pane shows the contents of the 'demo' folder. The title bar of the window reads 'C:\WINDOWS\system32\CMD.exe'. The command prompt shows the following commands and output:

```
E:\GitHub>cd demo
E:\GitHub\demo>dir
El volumen de la unidad E es DATA
El número de serie del volumen es: 32D9-6989

Directorio de E:\GitHub\demo

14/02/2016  11:56 a. m.  <DIR>          .
14/02/2016  11:56 a. m.  <DIR>          ..
14/02/2016  11:56 a. m.                201 .gitignore
14/02/2016  11:56 a. m.                38 README.md
                2 archivos            239 bytes
                2 dirs  669,954,220,032 bytes libres

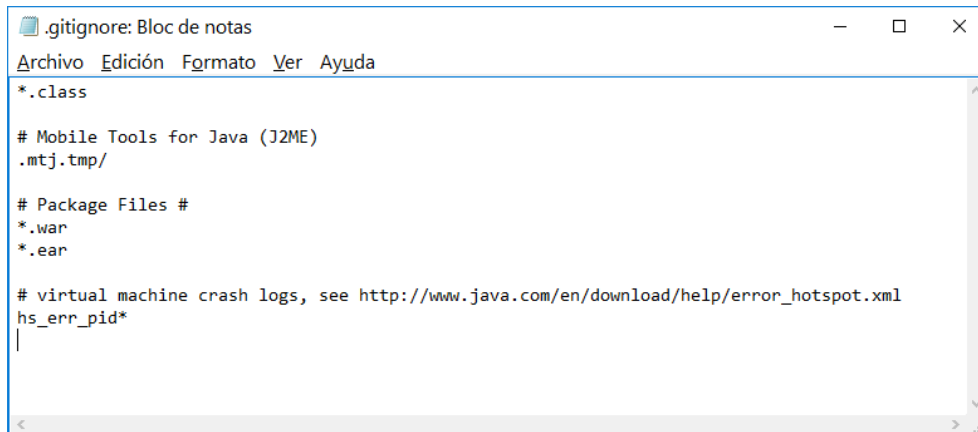
E:\GitHub\demo>
```



EDITAR REPOSITORIO

Modificar el archivo .gitignore

En el editor de texto carga el archivo `.gitignore` y elimina la línea que tenga `"*.jar"`, debe quedar como se muestra en la siguiente imagen:



```
.gitignore: Bloc de notas
Archivo Edición Formato Ver Ayuda

*.class

# Mobile Tools for Java (J2ME)
.mtj.tmp/

# Package Files #
*.war
*.ear

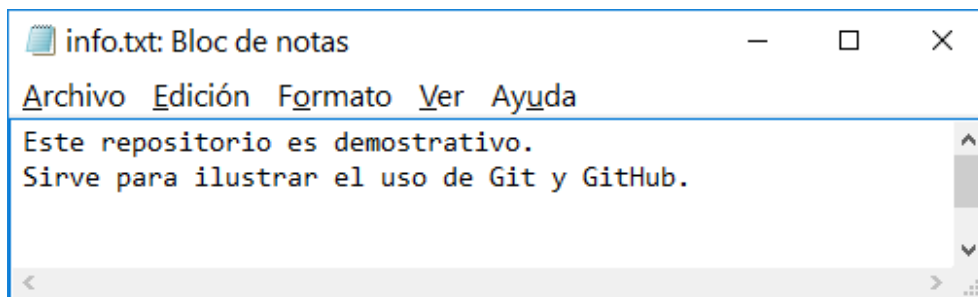
# virtual machine crash logs, see http://www.java.com/en/download/help/error_hotspot.xml
hs_err_pid*
|
```

Graba y cierra el archivo.

Crea un archivo

En la carpeta demo del repositorio procede a crear un archivo nuevo de nombre `info.txt`, y registra información sobre información que registrarás en tu repositorio.

La siguiente imagen muestra lo que podría ser el archivo `info.txt`.



```
info.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda

Este repositorio es demostrativo.
Sirve para ilustrar el uso de Git y GitHub.
```



ACTUALIZAR EL REPOSITORIO EN GITHUB

Verificar el estado del repositorio

El comando a ejecutar es:

```
git status
```

Para mi caso sería así:

```
E:\GitHub\demo>git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        info.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

El resultado nos indica que existen un archivo modificado y uno nuevo.

Confirmar los cambios en el repositorio local

Para confirmar los cambios, primero debes preparar la lista de archivos a confirmar con el comando **git add**.

Para mi caso, si quiero agregar todos los archivos a la lista:

```
E:\GitHub\demo>git add .
```

Para confirmar la lista de archivos preparada con **git add**, se debe utilizar el comando **git commit**.



Para mi caso, sería así:

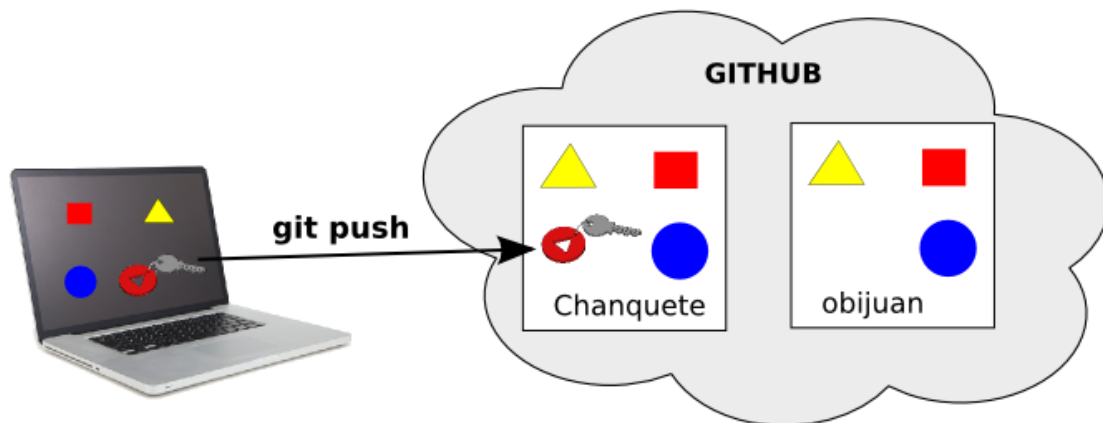
```
E:\GitHub\demo>git commit -m "Probando el repositorio."  
[master 5f781ff] Probando el repositorio.  
2 files changed, 1 deletion(-)  
create mode 100644 info.txt
```

Puedes utilizar nuevamente el comando `git status` para verificar el estado de tu repositorio.

```
E:\GitHub\demo>git status  
On branch master  
Your branch is ahead of 'origin/master' by 1 commit.  
(use "git push" to publish your local commits)  
nothing to commit, working directory clean
```

En este caso indica que el repositorio esta adelantado 1 commit con respecto a repositorio origen, y se debe utilizar `git push` para subir los cambios.

Subir los cambios a GitHub



Para subir los cambios a GitHub se utiliza el comando `git push`.

```
git push origin master
```

Cuando ejecutas este comando te solicita tu cuenta de usuario y clave, salvo que ya se encuentre configurado, como es mi caso.



A continuación se tiene el resultado para mi caso:

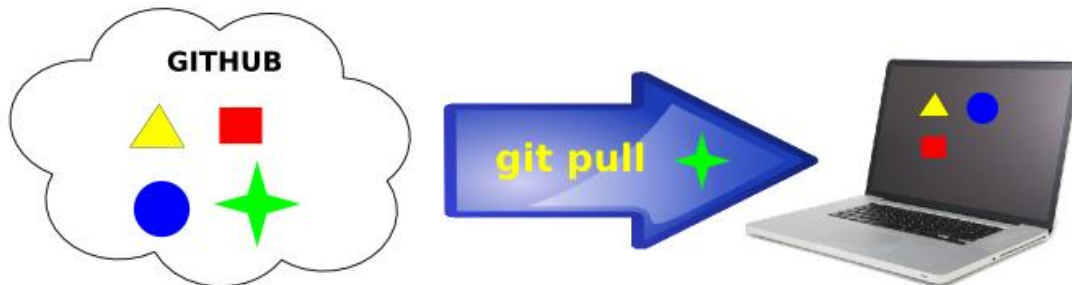
```
E:\GitHub\demo>git push origin master
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 371 bytes | 0 bytes/s, done.
Total 4 (delta 1), reused 0 (delta 0)
To https://github.com/gcoronelc/demo.git
    b1fc6e5..5f781ff  master -> master
```

Ahora puedes nuevamente verificar el estado de tu repositorio.

A continuación tienes el resultado para mi repositorio.

```
E:\GitHub\demo>git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

ACTUALIZAR EL REPOSITORIO LOCAL



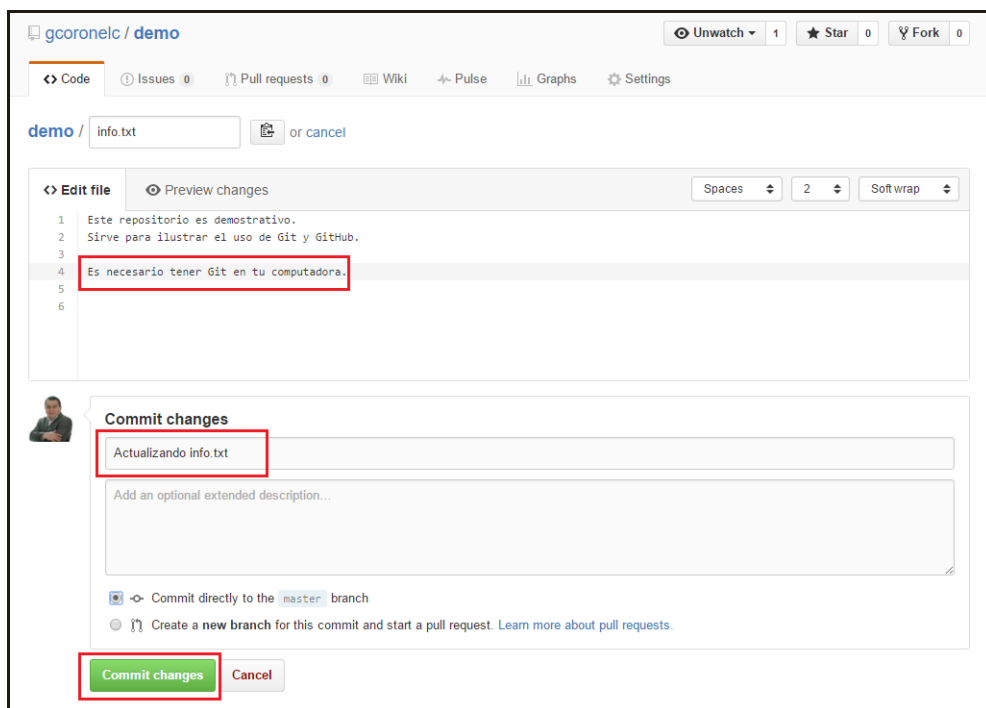
Puede suceder que el repositorio lo clones en la universidad y has realizado varias actualizaciones, luego quieres actualizar el repositorio de tú computadora en tu casa.

En estas situaciones debes usar el comando `git pull` para actualizar tu repositorio local.

Modifica el archivo `into.txt` en GitHub

Procede a editar el archivo `into.txt` en GitHub y agrégale una línea, y luego hazle `commit`.

A continuación tienes una imagen de como podrías realizarlo:





Actualiza tu repositorio local

Antes de que realices cualquier cambio en tu repositorio local, se recomienda que lo actualices con los últimos cambios, para eso debe utilizar el comando **git pull**.

Aquí tienes su sintaxis:

```
git pull origin master
```

Aquí tienes un ejemplo con mi repositorio:

```
E:\GitHub\demo>git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/gcoronelc/demo
* branch          master      -> FETCH_HEAD
   2a5a48e..2317f84 master      -> origin/master
Updating 2a5a48e..2317f84
Fast-forward
 info.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

El proceso se ha ejecutado correctamente, un archivo ha cambiado.

El archivo que ha cambiado es **info.txt**, he indica que una fila se ha insertado, pero, también una fila se ha eliminado, a continuación tienes el script para consultar el contenido del archivo **info.txt**:

```
E:\GitHub\demo>type info.txt
Este repositorio es demostrativo.
Sirve para ilustrar el uso de Git y GitHub.

Es necesario tener Git en tu computadora.
```

Esto ha sido una introducción a **Git** y **GitHub**, suficiente para utilizarlo como repositorio en los cursos de programación, pero si lo que necesitas es usarlo para control de versiones te recomiendo que consultes el material oficial de Git.

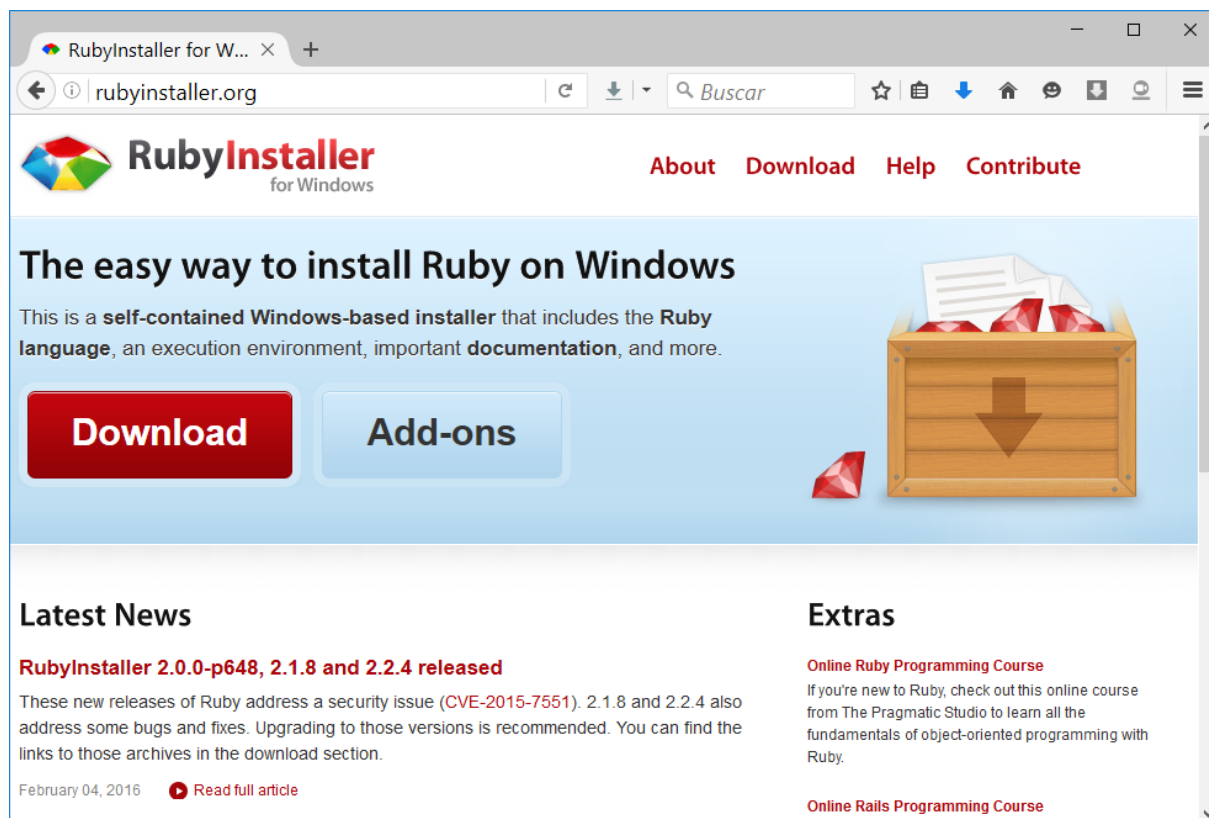


Capítulo 3 INSTALACIÓN DE RUBY

OBTENER SOFTWARE

Para obtener el software debe ir a la siguiente dirección:

<http://rubyinstaller.org/>



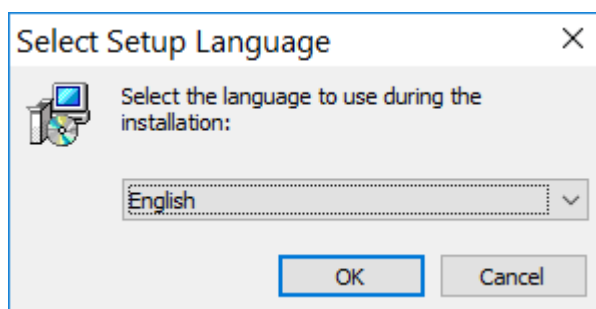


INSTALACIÓN

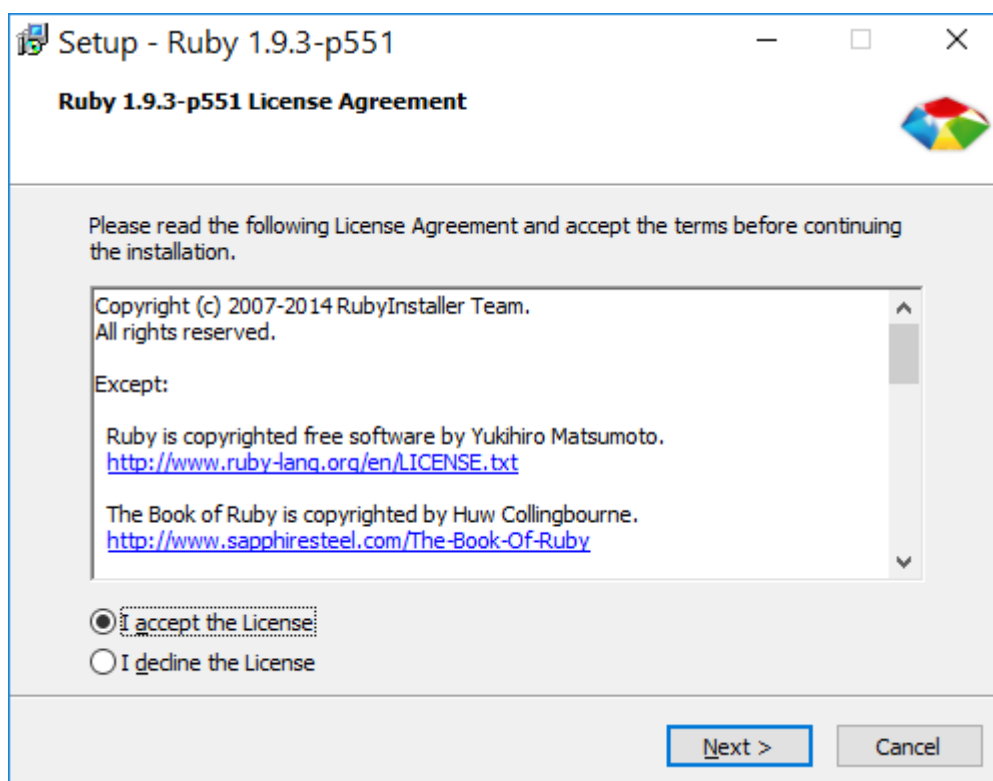
Ejecuta el programa que has descargado, por ejemplo, en mi caso el archivo es:

rubyinstaller-1.9.3-p551.exe

Como primera opción debes seleccionar el lenguaje:

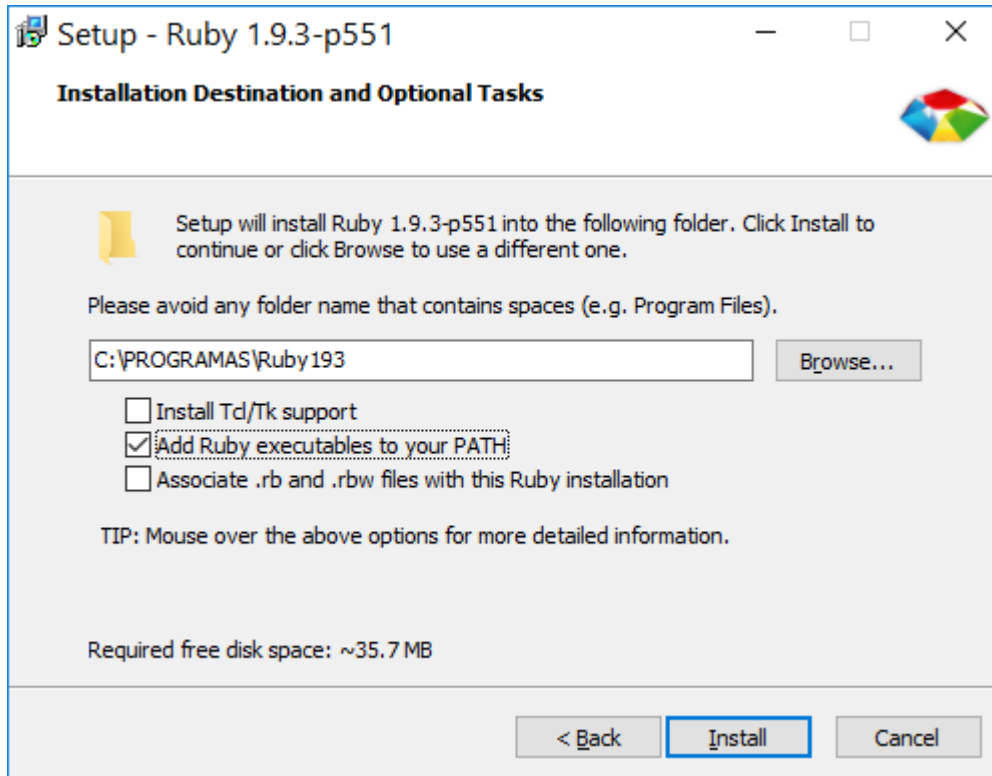


Debes aceptar los términos de licencia:





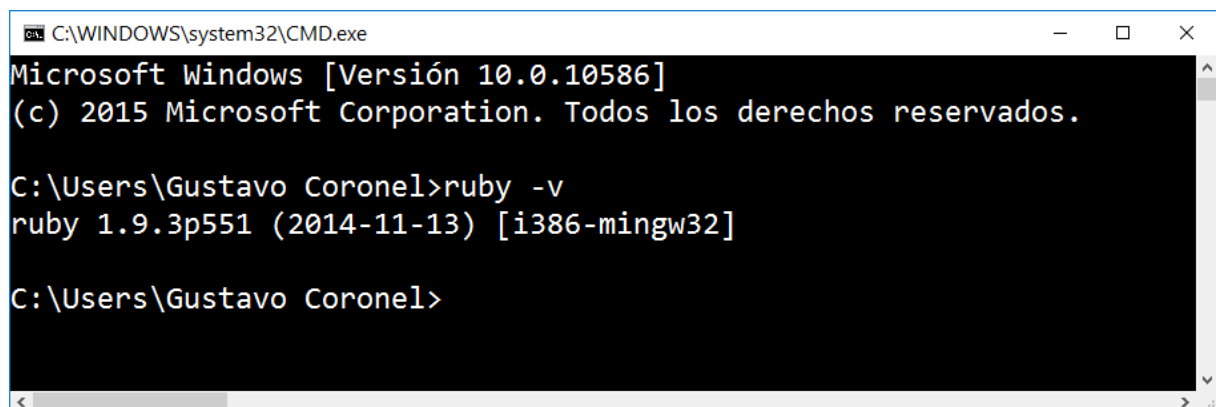
Luego debes elegir la carpeta donde se instalará y debes también elegir el check para agregar Ruby en la variable PATH de Windows, tal como lo puedes observar en la siguiente figura.



Finalmente haces click en el botón **Install** para proceder a instalarlo, no debe haber ningún problema.

Para verificar su instalación desde una consola de comandos ejecuta:

```
ruby -v
```





Capítulo 4

RHC CLIENT TOOLS

RHC

Las herramientas cliente de OpenShift, conocidos como RHC, están desarrolladas en Ruby. OpenShift se integra con el sistema de control de versiones **Git** para proporcionar un potente, control de versiones descentralizado para el código fuente de la aplicación.

INSTALACIÓN

Para su instalación debes utilizar el siguiente comando:

```
gem install rhc
```

Usa la consola para ejecutar el comando anterior, puedes hacerlo desde cualquier ubicación, no necesitas estar en un directorio en particular.



```
C:\WINDOWS\system32\CMD.exe

C:\Users\Gustavo Coronel>gem install rhc
Fetching: net-ssh-2.9.2.gem (100%)
Fetching: net-scp-1.2.1.gem (100%)
Fetching: net-ssh-gateway-1.2.0.gem (100%)
Fetching: net-ssh-multi-1.2.1.gem (100%)

.....

Installing RDoc documentation for archive-tar-minitar-0.5.2...
Installing RDoc documentation for highline-1.6.21...
Installing RDoc documentation for commander-4.2.1...
Installing RDoc documentation for httpclient-2.8.0...
Installing RDoc documentation for open4-1.3.4...
Installing RDoc documentation for rhc-1.38.4...

C:\Users\Gustavo Coronel>
```



CONFIGURACIÓN DE RHC

Comando a Utilizar

Para configurar RHC debes utilizar el siguiente comando:

```
rhc setup
```

Paso 1: Ejecutar Comando

```
C:\Users\Gustavo Coronel>rhc setup
```

Paso 2: Hostname del Servidor

En este caso vas a utilizar el que aparece por defecto, solo se presiona la tecla [Enter].

```
C:\Users\Gustavo Coronel>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| [Enter]

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

Paso 3: Inicio de Sesión

Debes usar tú correo electrónico y clave para iniciar sesión en OpenShift.

```
Login to openshift.redhat.com: gcoronelc@gmail.com| [Enter]
Password: *****| [Enter]
```



Paso 3: Generación de Token

El token te facilita el trabajo con las herramientas RHC, porque ya no te solicitará tu usuario y clave cada vez que ejecutas un comando.

```
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 [Enter]

Generating an authorization token for this client ... lasts about 1 month

Saving configuration to C:\Users\Gustavo Coronel\.openshift\express.conf
... done
```

Paso 4: Generación de Llaves SSH

Las llaves públicas SSH se utilizan para autenticar el trabajo con el repositorio git.

```
Your public SSH key must be uploaded to the OpenShift server to access
code. Upload now? (yes|no)
yes [Enter]

Since you do not have any keys associated with your OpenShift account,
your new key will be uploaded as the 'default' key.

Uploading key 'default' ... done

In order to fully interact with OpenShift you will need to install and
configure a git client if you have not already done so. Documentation for
installing other tools you will need for OpenShift can be found at
https://www.openshift.com/developers/install-the-client-tools
```



Paso 4: Recomendaciones Finales

Finalmente, la herramienta te hace recomendaciones para la creación de nuevas aplicaciones:

We recommend these free applications:

```
* Git for Windows - a basic git command line and GUI client http://msysgit.github.io/
* TortoiseGit - git client that integrates into the file explorer
http://code.google.com/p/tortoisegit/
Checking common problems
```

```
Your private SSH key file should be set as readable only to yourself. Please run
'chmod 600 C:\Users\Gustavo Coronel\.ssh\id_rsa'
```

```
Checking for a domain ... gcoronelc
```

```
Checking for applications ... none
```

```
Run 'rhc create-app' to create your first application.
```

Do-It-Yourself 0.1	rhc create-app <app name> diy-0.1
JBoss Application Server 7	rhc create-app <app name> jbossas-7
JBoss Data Virtualization 6	rhc create-app <app name> jboss-dv-6.1.0
JBoss Enterprise Application Platform	rhc create-app <app name> jbosseap-6
JBoss Unified Push Server 1.0.0.Beta1	rhc create-app <app name> jboss-unified-push-1
JBoss Unified Push Server 1.0.0.Beta2	rhc create-app <app name> jboss-unified-push-2
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
Vert.x 2.1	rhc create-app <app name> jboss-vertx-2.1
WildFly Application Server 10	rhc create-app <app name> jboss-wildfly-10
WildFly Application Server 8.2.1.Final	rhc create-app <app name> jboss-wildfly-8
WildFly Application Server 9	rhc create-app <app name> jboss-wildfly-9

```
You are using 0 of 3 total gears
```



```
The following gear sizes are available to you: small
```

```
Your client tools are now configured.
```

En estos momentos ya tienes tu cliente RHC configurado y listo para empezar a crear tus aplicaciones.



Capítulo 5

CREACIÓN DE UNA APLICACIÓN

CREACIÓN DE LA APLICACIÓN

Desde el portal de la OpenShift proceda a crear una aplicación.

Welcome to OpenShift

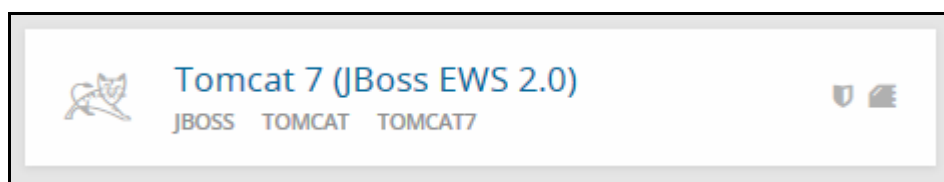
OpenShift helps you build and deploy web applications, mobile backends, service oriented architectures, and host your favorite services.

- 1. Choose a web framework or codebase to start from**
Try JBoss, PHP, Python, Ruby, Node.js, or create a new Drupal or Wordpress site instantly.
- 2. Add cartridges like MySQL or MongoDB to your application**
OpenShift lets you add services and tools to your application through **cartridges** - including databases, cache servers, management tools, and continuous integration servers.
- 3. Upload your code to OpenShift via Git**
Your source code is stored with your application in a Git version control repository.

[→ Create your first application now](#)

For more about OpenShift, visit the [OpenShift Developer Portal](#).

El tipo de aplicación debe ser Java con Apache Tomcat 7:



La URL pública asígnele app001:

Public URL	<input type="text" value="http://"/>	<input type="text" value="app001"/>	<input type="text" value="-gcoronelc.rhcloud.com"/>
OpenShift will automatically register this domain name for your application. You can add your own domain name later.			

Hacer click en el botón **Create Application**:

Create Application



Después que se crea la aplicación te presenta una página con algunas recomendaciones:

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://574abe8d2d5271767600000b@app001-gcoronelc.rhcloud.com/~/.git/  
/app001.git/  
cd app001/
```

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](#).

A lo que debes prestar atención es al comando para que clones tú repositorio, a continuación tienes un ejemplo:

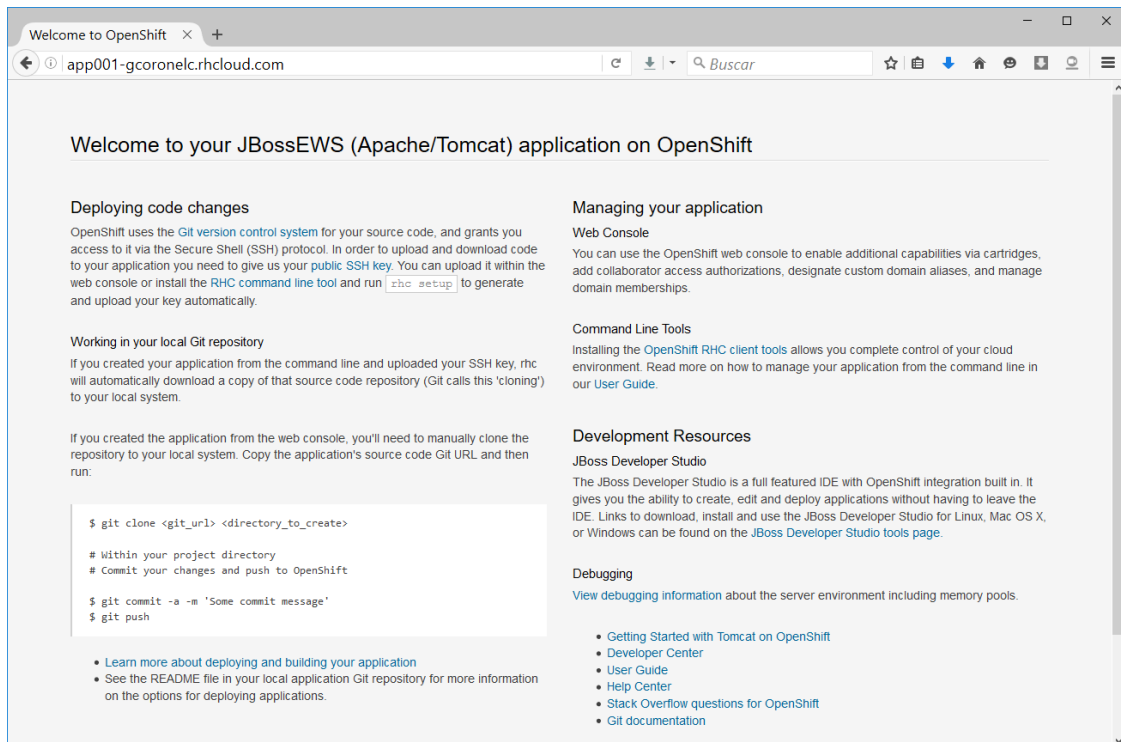
```
git clone ssh://574abe8d2d5271767600000b@app001-gcoronelc.rhcloud.com/~/.git/app001.git  
cd app001/
```

Y el comando para que subas tus cambios, a continuación tienes un ejemplo:

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



Por defecto tu aplicación tiene el siguiente contenido:



Welcome to your JBossEWS (Apache/Tomcat) application on OpenShift

Deploying code changes

OpenShift uses the [Git version control system](#) for your source code, and grants you access to it via the Secure Shell (SSH) protocol. In order to upload and download code to your application you need to give us your [public SSH key](#). You can upload it within the web console or install the [RHC command line tool](#) and run `rhc setup` to generate and upload your key automatically.

Managing your application

Web Console

You can use the OpenShift web console to enable additional capabilities via cartridges, add collaborator access authorizations, designate custom domain aliases, and manage domain memberships.

Command Line Tools

Installing the [OpenShift RHC client tools](#) allows you complete control of your cloud environment. Read more on how to manage your application from the command line in our [User Guide](#).

Working in your local Git repository

If you created your application from the command line and uploaded your SSH key, rhc will automatically download a copy of that source code repository (Git calls this 'cloning') to your local system.

If you created the application from the web console, you'll need to manually clone the repository to your local system. Copy the application's source code Git URL and then run:

```
$ git clone <git_url> <directory_to_create>

# Within your project directory
# Commit your changes and push to OpenShift

$ git commit -a -m 'Some commit message'
$ git push
```

Development Resources

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](#).

Debugging

[View debugging information](#) about the server environment including memory pools.

- [Getting Started with Tomcat on OpenShift](#)
- [Developer Center](#)
- [User Guide](#)
- [Help Center](#)
- [Stack Overflow questions for OpenShift](#)
- [Git documentation](#)

- [Learn more about deploying and building your application](#)
- [See the README file in your local application Git repository for more information on the options for deploying applications.](#)

PUBLICAR UN WAR

1. Crea una aplicación web y procede a compilarla, por ejemplo: **demo001.war**
2. Borra el contenido por defecto:



```
C:\WINDOWS\system32\CMD.exe

E:\Java\OpenShift>cd app001

E:\Java\OpenShift\app001>git rm -rf src/ pom.xml
rm 'pom.xml'
rm 'src/main/java/.gitkeep'
rm 'src/main/resources/.gitkeep'
rm 'src/main/webapp/WEB-INF/web.xml'
rm 'src/main/webapp/images/jbosscorp_logo.png'
rm 'src/main/webapp/index.html'
rm 'src/main/webapp/snoop.jsp'

E:\Java\OpenShift\app001>git commit -m "borrando contenido por defecto"
[master 1744a3a] borrando contenido por defecto
7 files changed, 657 deletions(-)
delete mode 100755 pom.xml
delete mode 100644 src/main/java/.gitkeep
delete mode 100644 src/main/resources/.gitkeep
delete mode 100755 src/main/webapp/WEB-INF/web.xml
delete mode 100644 src/main/webapp/images/jbosscorp_logo.png
delete mode 100755 src/main/webapp/index.html
delete mode 100644 src/main/webapp/snoop.jsp

E:\Java\OpenShift\app001>
```

3. Añadir el archivo WAR al directorio **webapps** y actualizar el repositorio en la nube.

```
C:\WINDOWS\system32\CMD.exe

E:\Java\OpenShift\app001>git add .

E:\Java\OpenShift\app001>git commit -m "Nueva aplicación"
[master 6616cad] Nueva aplicación
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 webapps/demo001.war

E:\Java\OpenShift\app001>git push

.....
.....

E:\Java\OpenShift\app001>
```

4. Reiniciar la aplicación.

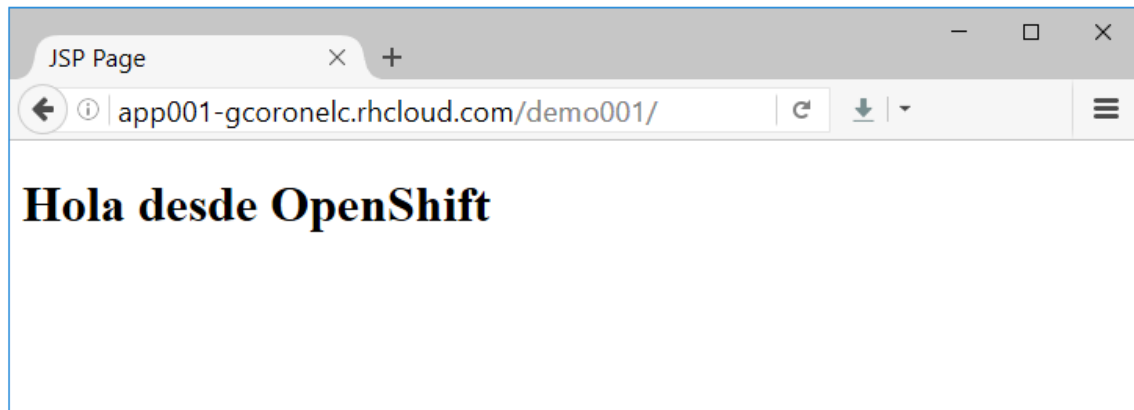


```
C:\WINDOWS\system32\CMD.exe

E:\Java\OpenShift\app001>rhc app restart -a app001
RESULT:
app001 restarted

E:\Java\OpenShift\app001>
```

5. Ejecuta la aplicación.





Referencias

- <https://blog.openshift.com/how-to-install-the-openshift-rhc-client-tools-on-windows/>
- <https://developers.openshift.com/getting-started/>
- <https://developers.openshift.com/getting-started/windows.html>