**Manual de Instalación y Configuración Git**

Contenido

[1. Instalación y configuración servidor Git 3](#_Toc367810624)

[1.1. Instalando desde código fuente 4](#_Toc367810627)

[1.2. Instalando desde Plataforma 5](#_Toc367810628)

[1.2.1. Instalación en Linux 5](#_Toc367810629)

[1.2.2. Instalación en Mac 6](#_Toc367810630)

[1.2.3. Instalación en Windows 6](#_Toc367810631)

[1.3. Configuración y creación de repositorio de Git 6](#_Toc367810632)

[2. Configuración cliente 10](#_Toc367810637)

[2.1. Configuracion cliente Linux-Mac 10](#_Toc367810638)

[2.2. Configuración cliente Windows 12](#_Toc367810639)

[3. Referencias 14](#_Toc367810640)

Índice de Figuras

Figura 1 – Home Principal Git 3

Figura 2 – Pagina de descarga server por Plataforma 4

Figura 3 – Home GitHub 6

Figura 4 – Creacion de cuenta GitHub 7

Figura 5 – Manejo de repositorios en GitHub 8

Figura 6 – Creación de repositorio en GitHub 8

Figura 7 – Home GitHub for Windows 12

Figura 8 – Inicio de sesión con GitHub for Windows 13

Figura 9 – Listado de repositorios en GitHub for Windows 13

# Instalación y configuración servidor Git

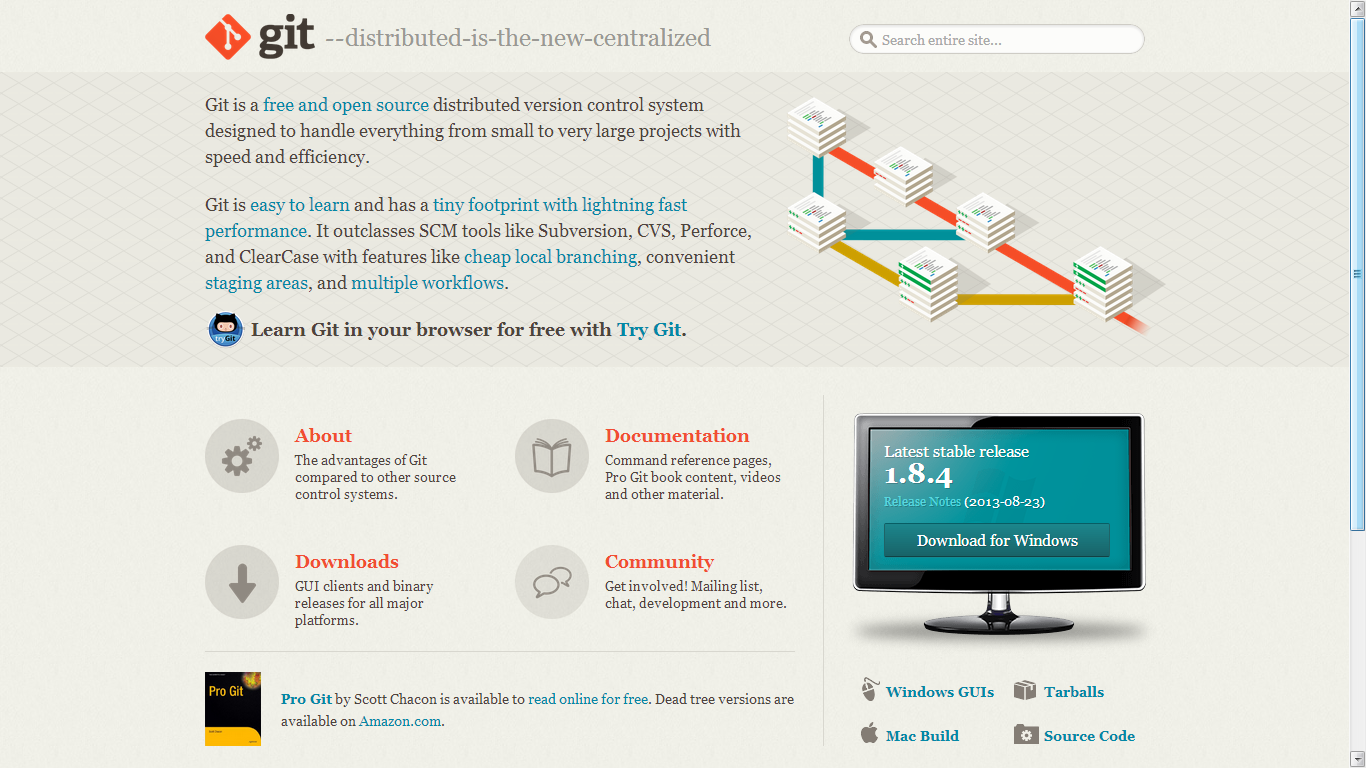
Git es software libre bajo la licencia GNU/GPL v.2, y su lado más conocido es a través de GitHub, una poderosa y versátil aplicación web que permite una administración intuitiva de los repositorios y provee servicios complementarios como visualizaciones del código, manejo de incidencias, estadísticas, wiki entre otras.

Git es un software que permite el control de versiones, no solo de software sino de todo tipo de archivos.

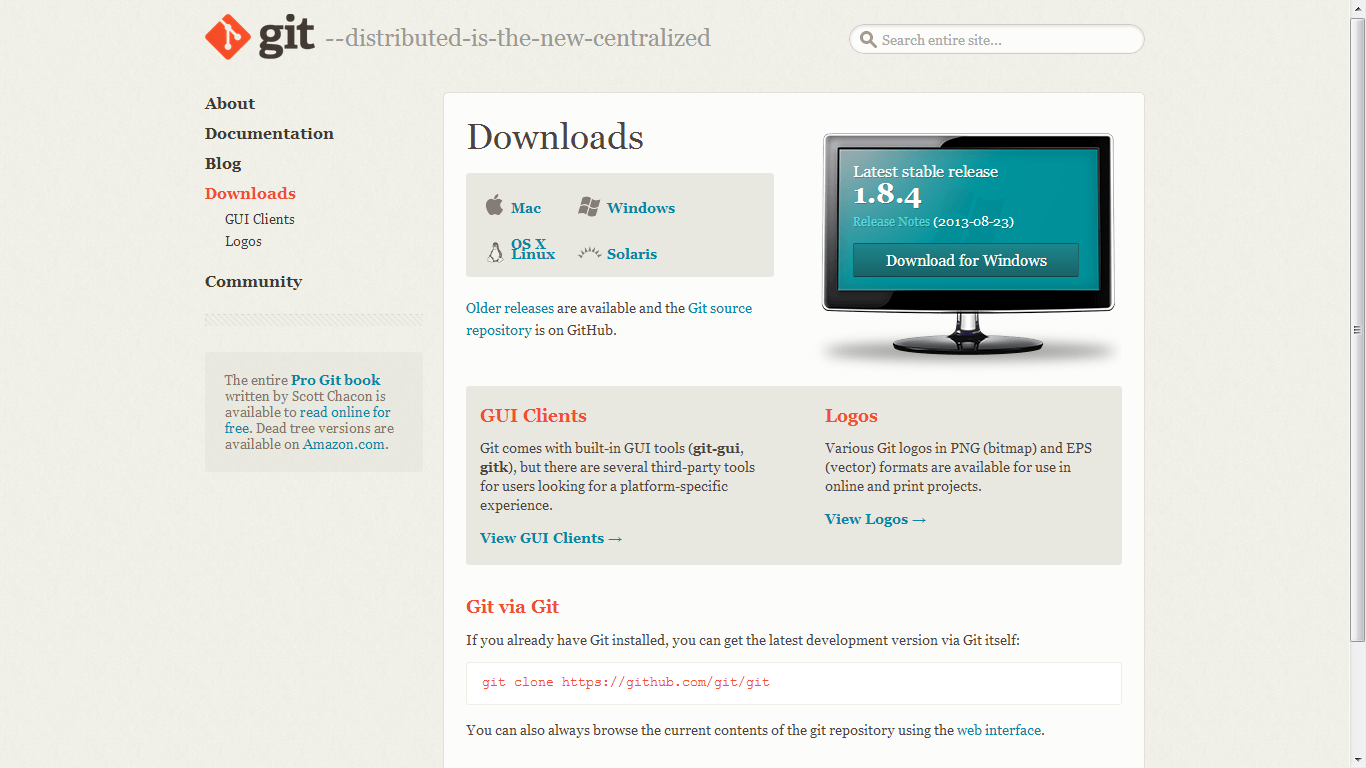
Las dos principales formas de instalar Git son a través del código fuente, o instalar un paquete existente para una plataforma específica.

Git es una herramienta de control de revisiones confiable, versátil y multipropósito.

La página de donde se puede descargar el código fuente para la instalación del servidor es [http://git-scm.com](http://git-scm.com/)



## Figura 1 – Home Principal Git



## Figura 2 – Pagina de descarga server por Plataforma

# Instalando desde código fuente

Si puedes, en general es útil instalar Git desde código fuente, porque obtendrás la versión más reciente. Cada versión de Git tiende a incluir útiles mejoras en la interfaz de usuario, por lo que utilizar la última versión es a menudo el camino más adecuado si te sientes cómodo compilando software desde código fuente. También ocurre que muchas distribuciones de Linux contienen paquetes muy antiguos; así que a menos que estés en una distribución muy actualizada o estés usando backports, instalar desde código fuente puede ser la mejor apuesta.

Para instalar Git, necesitas tener las siguientes librerías de las que Git depende: curl, zlib, openssl, expat, y libiconv. Por ejemplo, si estás en un sistema que tiene yum (como Fedora) o apt-get (como un sistema basado en Debian), puedes usar uno de estos comandos para instalar todas las dependencias:

$ yum install curl-devel expat-devel gettext-devel \openssl-devel zlib-devel

$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \ libz-dev

Cuando tengas todas las dependencias necesarias, puedes descargar la versión más reciente de Git desde su página web:

<http://git-scm.com/download>

Luego compila e instala:

$ tar -zxf git-1.6.0.5.tar.gz

$ cd git-1.6.0.5

$ make prefix=/usr/local all

$ sudo make prefix=/usr/local install

Una vez hecho esto, también puedes obtener Git a través del propio Git para futuras actualizaciones:

$ git clone git://git.kernel.org/pub/scm/git/git.git

# Instalando desde Plataforma

# Instalación en Linux

Si quieres instalar Git en Linux a través de un instalador binario, en general puedes hacerlo a través de la herramienta básica de gestión de paquetes que trae tu distribución. Si estás en Fedora, puedes usar yum:

$ yum install git-core

O si estás en una distribución basada en Debian como Ubuntu, prueba con apt-get:

$ apt-get install git

# Instalación en Mac

Hay dos maneras fáciles de instalar Git en un Mac. La más sencilla es usar el instalador gráfico de Git, que puedes descargar desde la página de Google Code : <http://code.google.com/p/git-osx-installer>

La otra manera es instalar Git a través de MacPorts (http://www.macports.org). Si tienes MacPorts instalado, instala Git con:

$ sudo port install git-core +svn +doc +bash\_completion +gitweb

No necesitas añadir todos los extras, pero probablemente quieras incluir +svn en caso de que alguna vez necesites usar Git con repositorios Subversion.

# Instalación en Windows

Instalar Git en Windows es muy fácil. El proyecto msysGit tiene uno de los procesos de instalación más sencillos. Simplemente descarga el archivo exe del instalador desde la página de GitHub, y ejecútalo:

<http://msysgit.github.com/>

Una vez instalado, tendrás tanto la versión de línea de comandos (incluido un cliente SSH que nos será útil más adelante) como la interfaz gráfica de usuario estándar.

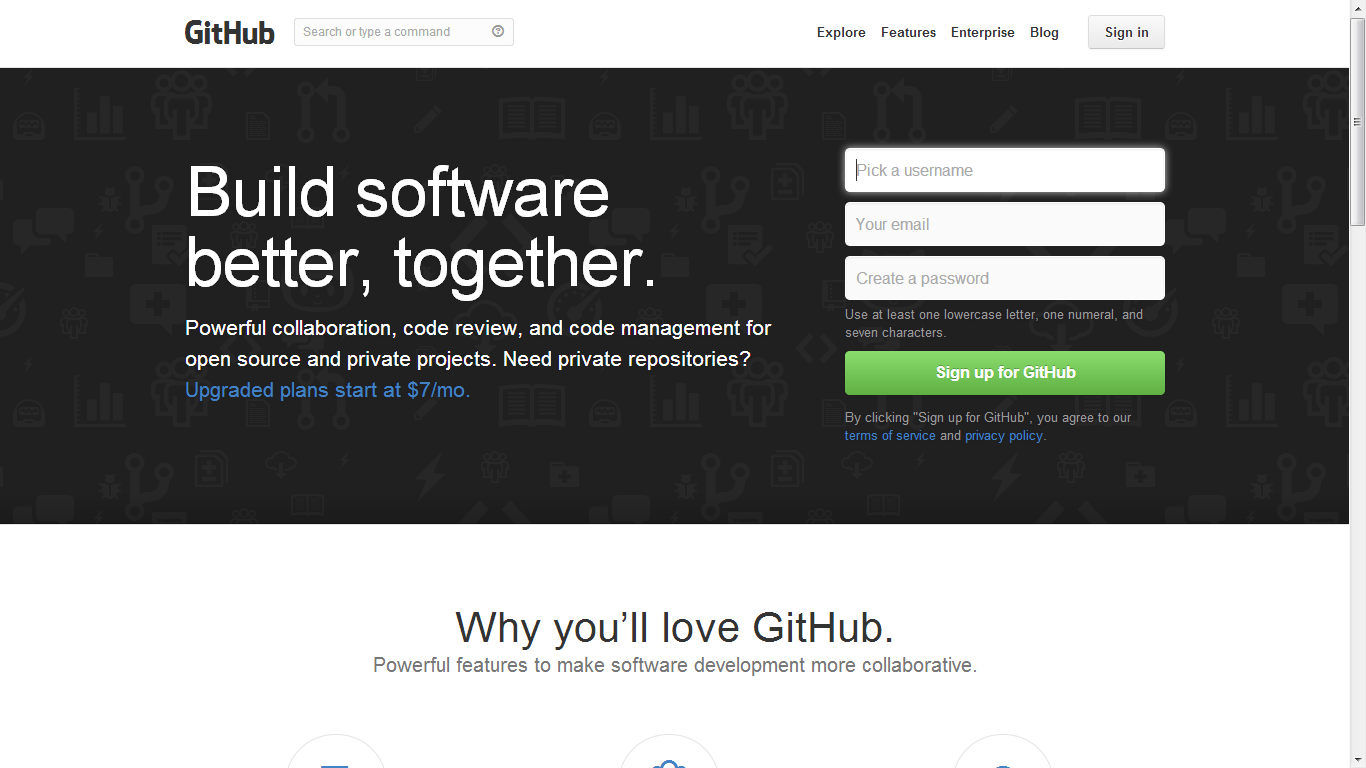
# Configuración y creación de repositorio de Git

Para la creación de nuestro repositorio central, ingresaremos a la página <http://github.com> que será quien nos dé un entorno público donde almacenaremos nuestros archivos, para ello debemos seguir los siguientes pasos:

**Crear una cuenta.**

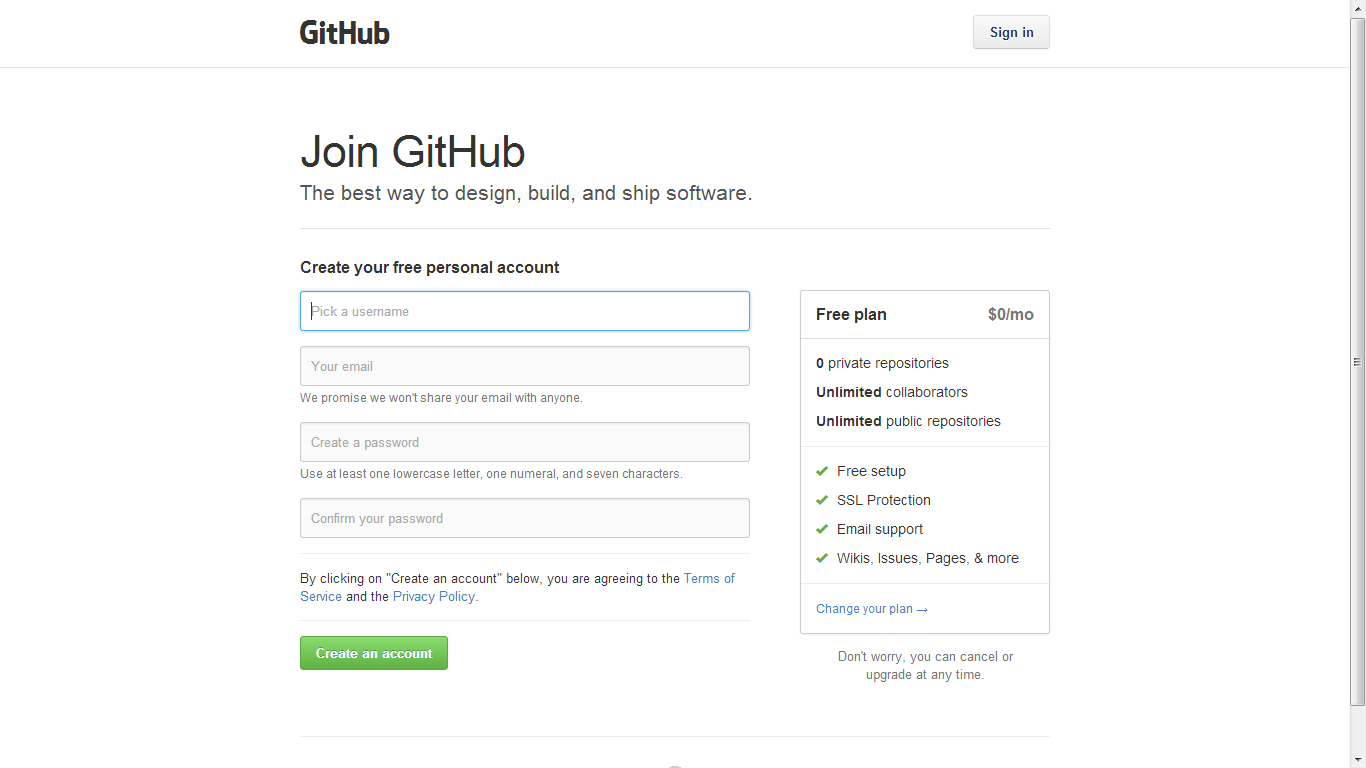
Para crear una cuenta GitHub vamos a seguir los siguientes pasos:

1. Entra a [GitHub](https://www.github.com/).



## Figura 3 – Home GitHub

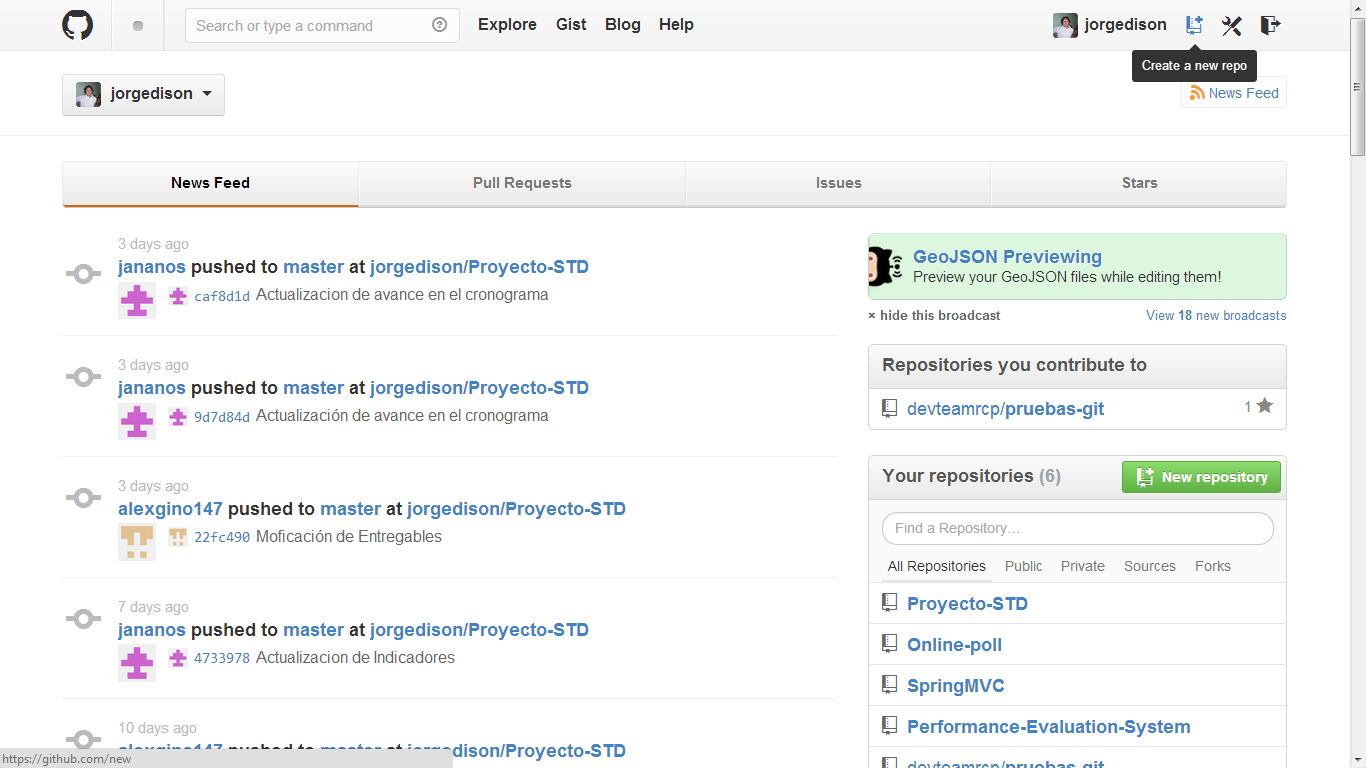
1. Haz clic en Sign up for GitHub o ingresar a <https://github.com/signup/free>
2. Ahora en Create a free account, para una cuenta gratuita.
3. Finalmente, solo habrá que llenar un pequeño formulario con los siguientes datos:
   * Nombre de usuario.
   * Dirección email.
   * Contraseña.
   * Confirmar contraseña.



## Figura 4 – Creación de cuenta GitHub

**Crear repositorios**

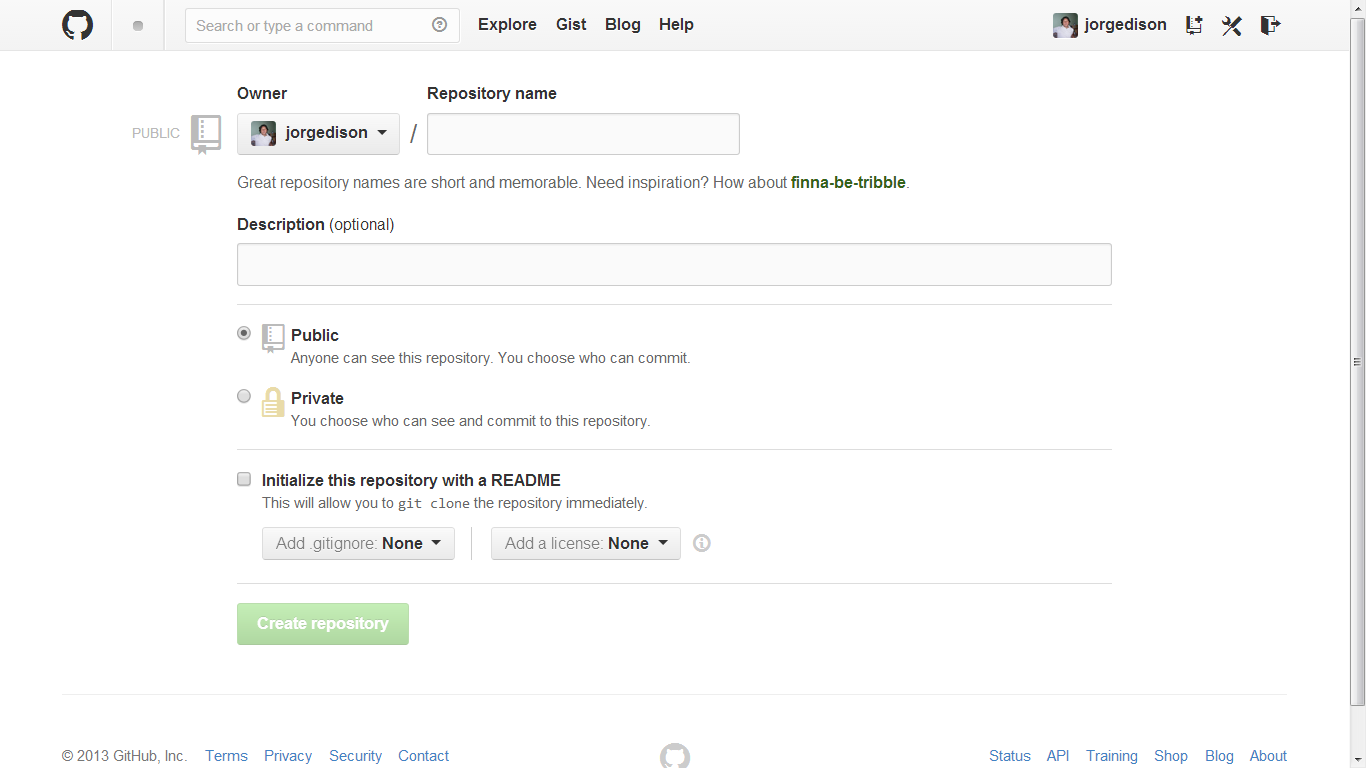
Para crear un repositorio en GitHub, solo hay que seleccionar el botón “Create a New Repo”, de la barra de herramientas, habiendo entrado a [GitHub](https://www.github.com/)con tu cuenta:



## Figura 5 – Manejo de repositorios en GitHub

Ahora habrá que llenar dos datos:

1. Nombre del repositorio
2. Descripción del repositorio (opcional)



## Figura 6 – Creación de repositorio en GitHub

**Crear un proyecto**

Al crear el repositorio, inmediatamente nos va a llevar a él. Como nuestro proyecto no tiene nada en su interior, no nos va a mostrar más que una ayuda para subir archivos y proyectos.

Para crear un proyecto desde cero, habrá que comenzar creando los archivos del mismo y luego subiéndolos a la página.

En el primer recuadro de la ayuda, verás una serie de comandos para el terminal.

touch README.md

git init

git add README.md

git commit -m "comentario"

git remote add origin https://github.com/jorgedison/Repositorio.git

git push -u origin master

Antes de seguirlos, tendrás que instalar git:

sudo apt-get install git

**Subir proyecto**

Para subir un proyecto ya realizado a GitHub, habrá que seguir exactamente los mismos pasos, ya que para iniciar un proyecto nuevo, habrá que subir los archivos creados y luego modificarlos en el editor de la página.

En este caso, los archivos junto con su contenido, ya están hechos. Solo hay que subirlos y editarlos si surge algún inconveniente.

Para subir un archivo hay que ubicarse en la carpeta del repositorio y seguir estos pasos:

git add archivo

git commit -m "comentario"

git push

Es importante realizar los 3, ya que si no se ingresa un comentario, no se realiza el cambio.

# Configuración cliente

# Configuracion cliente Linux-Mac

Ahora que tienes Git en tu sistema, querrás hacer algunas cosas para personalizar tu entorno de Git. Deberías tener que hacer estas cosas sólo una vez; se mantendrán entre actualizaciones. También puedes cambiarlas en cualquier momento volviendo a ejecutar los comandos correspondientes.

Git trae una herramienta llamada git config que te permite obtener y establecer variables de configuración que controlan el aspecto y funcionamiento de Git. Estas variables pueden almacenarse en tres sitios distintos:

* Archivo /etc/gitconfig: Contiene valores para todos los usuarios del sistema y todos sus repositorios. Si pasas la opción --system a git config, lee y escribe específicamente en este archivo.
* Archivo ~/.gitconfig file: Específico a tu usuario. Puedes hacer que Git lea y escriba específicamente en este archivo pasando la opción --global.
* Archivo config en el directorio de git (es decir, .git/config) del repositorio que estés utilizando actualmente: Específico a ese repositorio. Cada nivel sobrescribe los valores del nivel anterior, por lo que los valores de .git/config tienen preferencia sobre los de /etc/gitconfig.

En sistemas Windows, Git busca el archivo .gitconfig en el directorio $HOME (C:\Documents and Settings\$USER para la mayoría de usuarios). También busca en el directorio /etc/gitconfig, aunque esta ruta es relativa a la raíz MSys, que es donde quiera que decidieses instalar Git en tu sistema Windows cuando ejecutaste el instalador.

**Tu identidad**

Lo primero que deberías hacer cuando instalas Git es establecer tu nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información, y es introducida de manera inmutable en los commits que envías:

$ git config --global user.name " Jorge Rodriguez"

$ git config --global user.email [jrodriguez@rcp.pe](mailto:jrodriguez@rcp.pe)

De nuevo, sólo necesitas hacer esto una vez si especificas la opción --global, ya que Git siempre usará esta información para todo lo que hagas en ese sistema. Si quieres sobrescribir esta información con otro nombre o dirección de correo para proyectos específicos, puedes ejecutar el comando sin la opción --global cuando estés en ese proyecto.

**Tu editor**

Ahora que tu identidad está configurada, puedes elegir el editor de texto por defecto que se utilizará cuando Git necesite que introduzcas un mensaje. Si no indicas nada, Git usa el editor por defecto de tu sistema, que generalmente es Vi o Vim. Si quieres usar otro editor de texto, como Emacs, puedes hacer lo siguiente:

$ git config --global core.editor emacs

Tu herramienta de diferencias

Otra opción útil que puede que quieras configurar es la herramienta de diferencias por defecto, usada para resolver conflictos de unión (merge). Digamos que quieres usar vimdiff:

$ git config --global merge.tool vimdiff

Git acepta kdiff3, tkdiff, meld, xxdiff, emerge, vimdiff, gvimdiff, ecmerge, y opendiff como herramientas válidas. También puedes configurar la herramienta que tú quieras; véase el Capítulo 7 para más información sobre cómo hacerlo.

Comprobando tu configuración

Si quieres comprobar tu configuración, puedes usar el comando git config --list para listar todas las propiedades que Git ha podido encontrar en ese punto:

$ git config --list

user.name=Scott Chacon

user.email=schacon@gmail.com

color.status=auto

color.branch=auto

color.interactive=auto

color.diff=auto

Puede que veas claves repetidas, porque Git lee la misma clave de distintos archivos (/etc/gitconfig y ~/.gitconfig, por ejemplo). En ese caso, Git usa el último valor para cada clave única que ve.

También puedes comprobar qué valor cree Git que tiene una clave específica ejecutando git config {clave}:

$ git config user.name

Jorge Rodríguez

# Configuración cliente Windows

Para el uso de nuestra herramienta cliente dentro de entorno Windows, usaremos GitHub for Windows para lo cual debemos seguir los siguientes pasos:

1. Ingresar a la siguiente página <http://windows.github.com/>
2. Descargar el instalador de GitHub for Windows

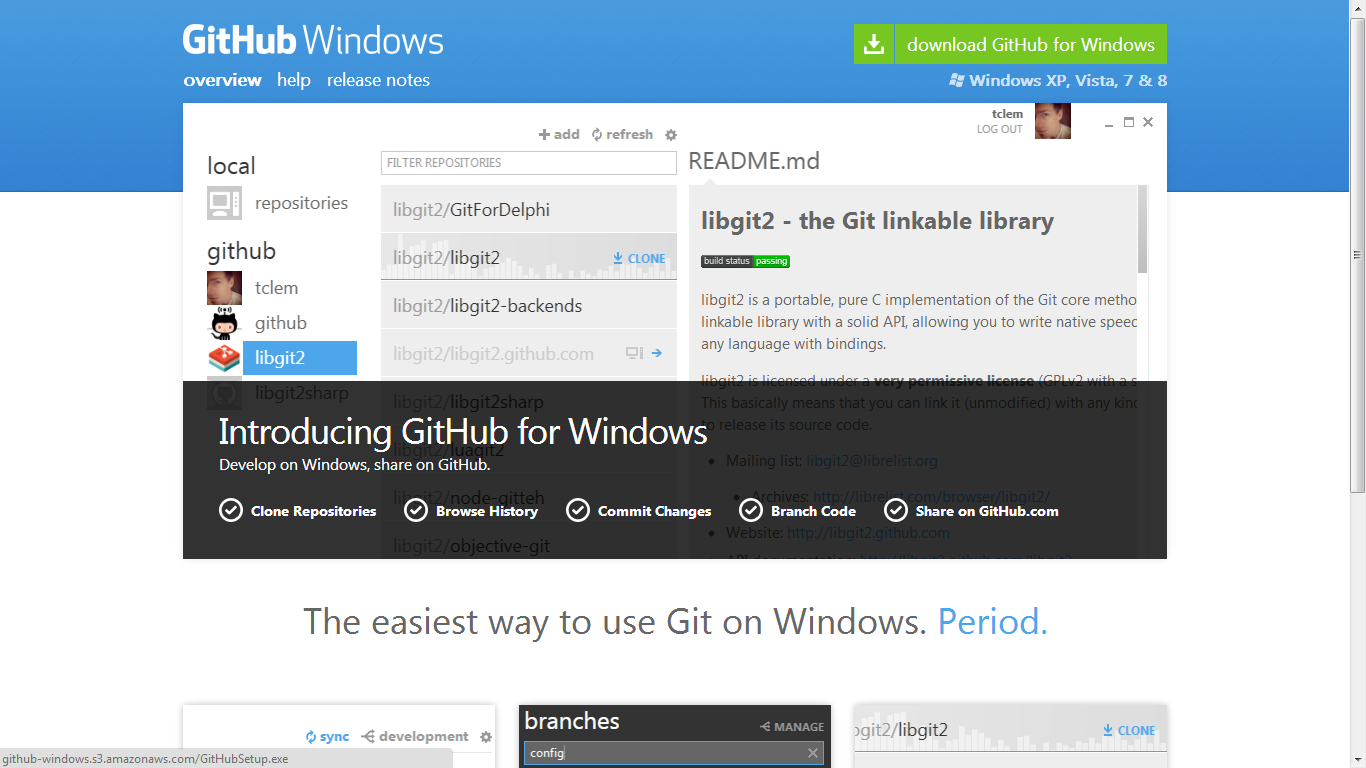


Figura 7 – Home GitHub for Windows

1. Ejecutamos e instalaremos el archivo descargado GitHubSetup.exe
2. Una vez instalado, iniciamos el ejecutable y se mostrara una interfaz parecida a la siguiente:

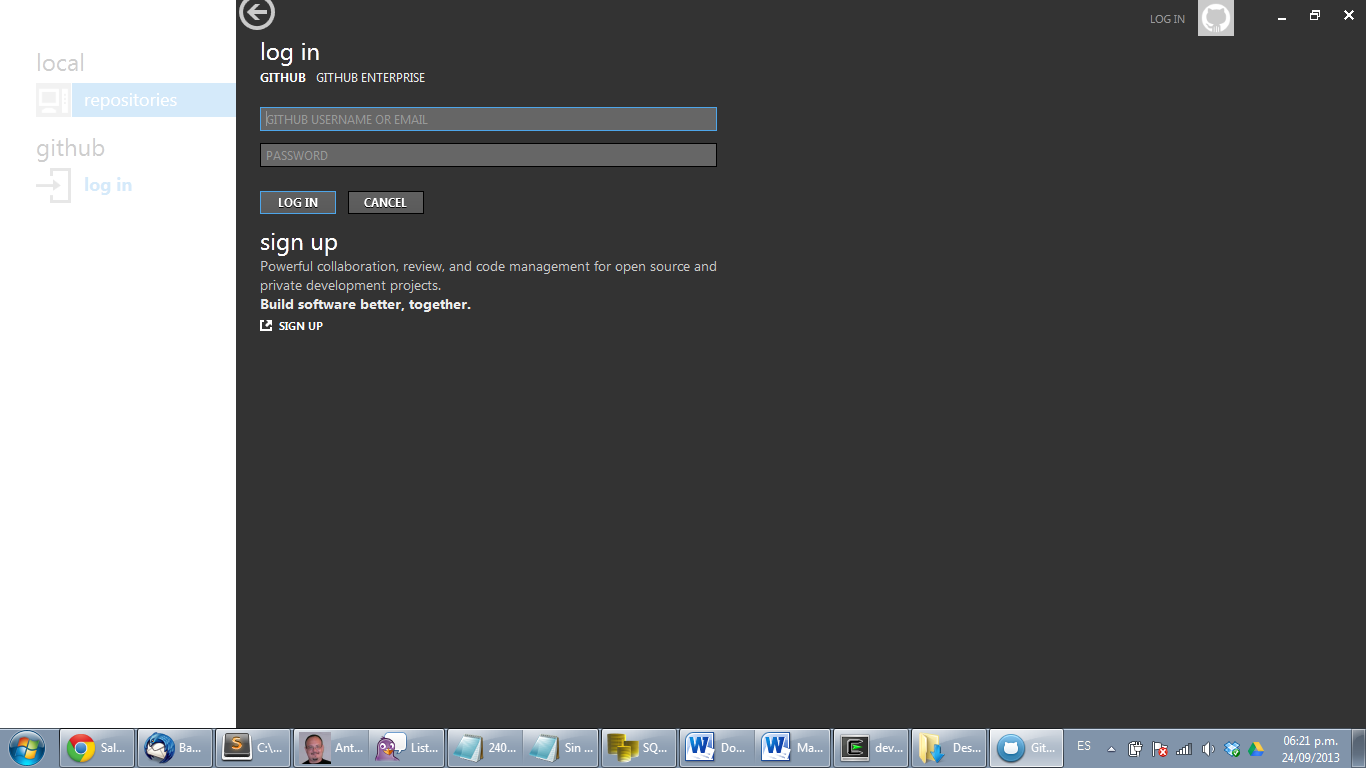


Figura 8 – Inicio de sesión con GitHub for Windows

1. Ingresamos nuestro usuario y contraseña que creamos en <http://github.com> e iniciamos sesión, la interfaz nos mostrara una interfaz con los repositorios que tenemos creados o donde tenemos participación.

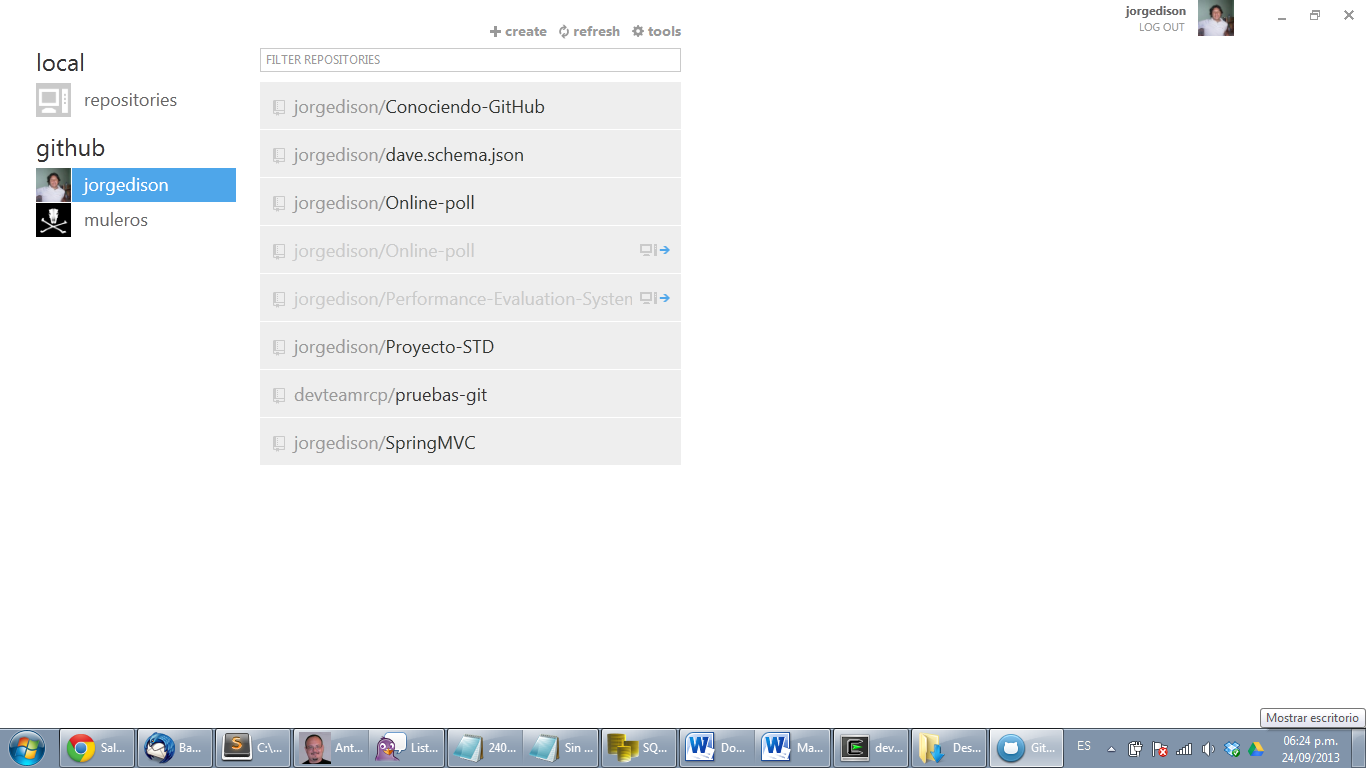


Figura 9 – Listado de repositorios en GitHub for Windows

# Referencias

* [http://git-scm.com](http://git-scm.com/)
* <http://rogerdudler.github.io/git-guide/index.es.html>
* <http://www.edy.es/dev/docs/git-guia-rapida/>