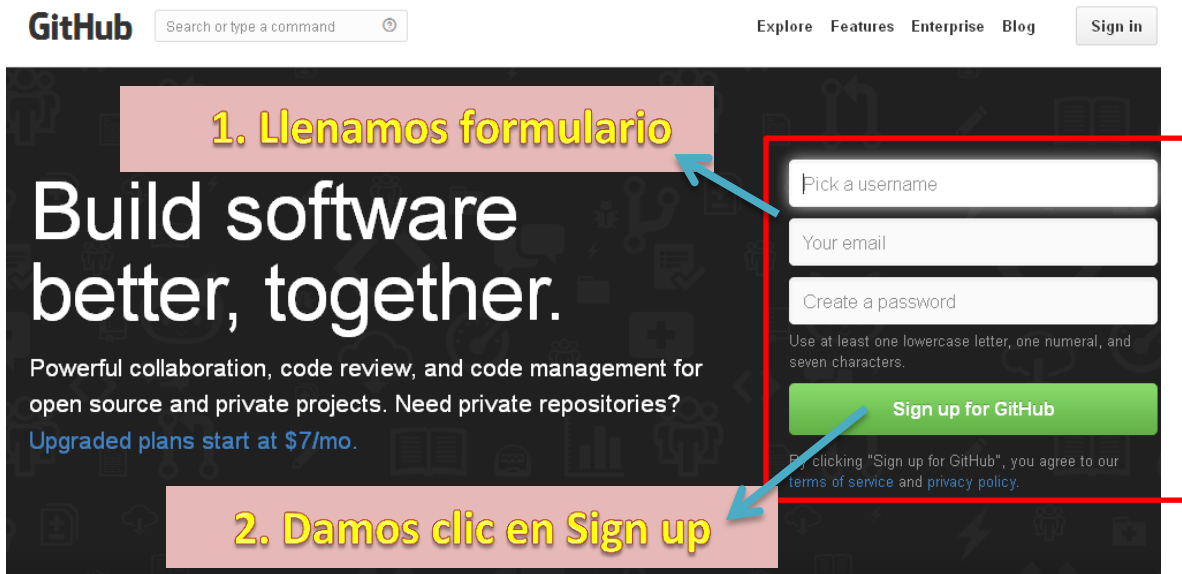


Manual para la utilización de git | hub

Lo primero para empezar a utilizar esta herramienta que nos permitirá trabajar nuestros proyectos por versiones, es registrarnos en github.com de forma gratuita, llenamos los espacios que nos piden y damos clic en sign up



The image shows the GitHub homepage with a sign-up form highlighted by a red rectangle. Two yellow text boxes with blue arrows provide instructions: '1. Llenamos formulario' points to the input fields, and '2. Damos clic en Sign up' points to the 'Sign up for GitHub' button.

1. Llenamos formulario

2. Damos clic en Sign up

GitHub Search or type a command

Explore Features Enterprise Blog Sign in

Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Need private repositories?
[Upgraded plans start at \\$7/mo.](#)

Pick a username

Your email

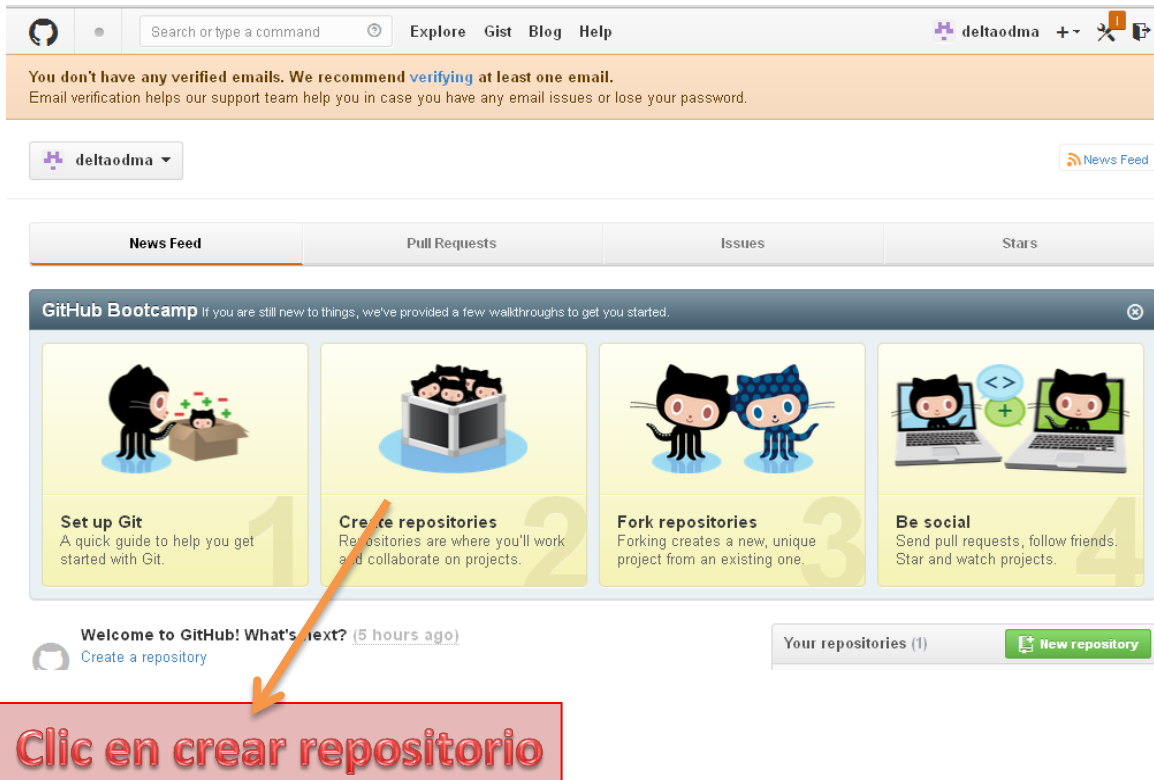
Create a password

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

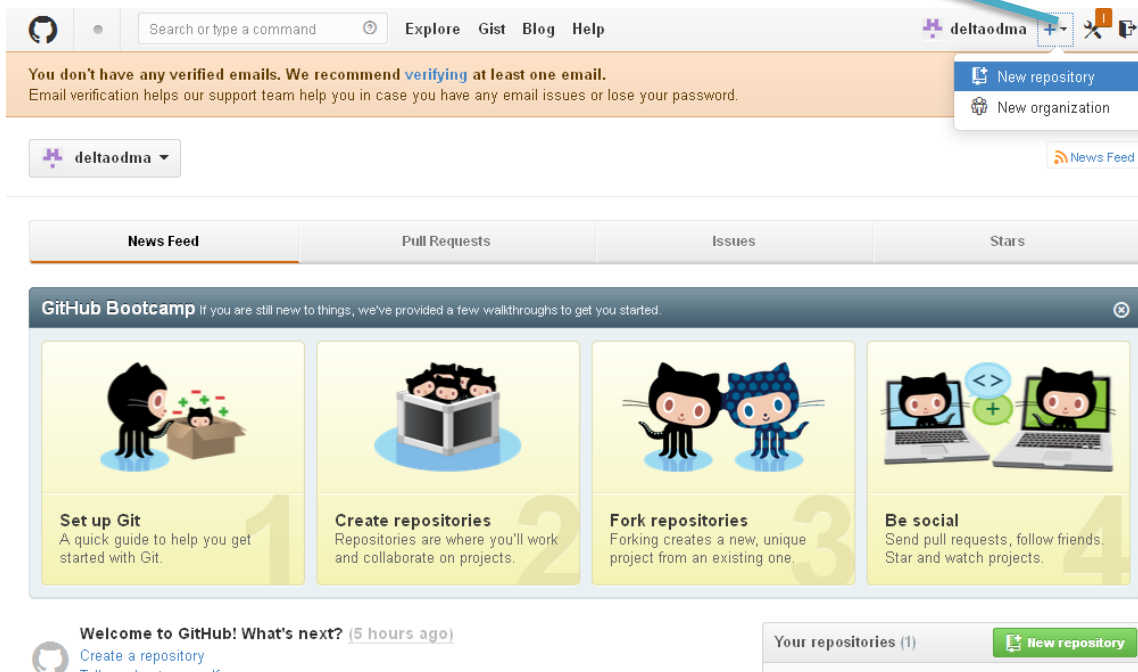
Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#).

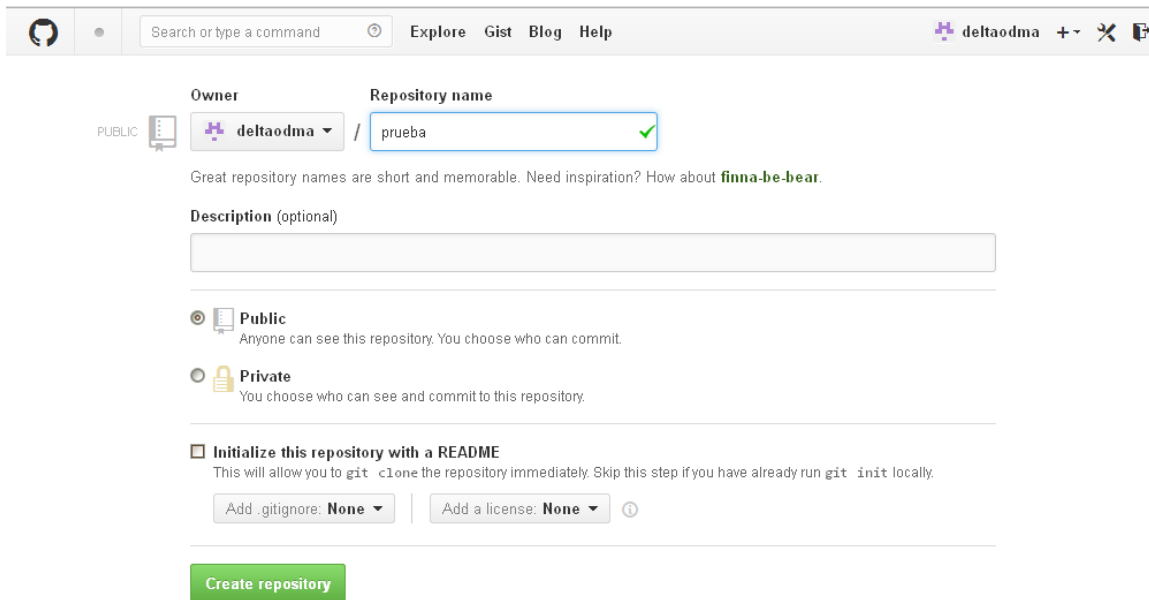
Cuando tengamos nuestra cuenta accedemos y creamos un repositorio, el cual va a hacer de carpeta donde vamos a ir trabajando nuestro proyecto.



Después de estar logueados nos va a salir el primer pantallazo de github nos vamos al icono de un mas (+) ubicado en la parte de arriba a la derecha en al dar clic nos saldrá dos opciones vamos a seleccionar la de crear nuevo repositorio.



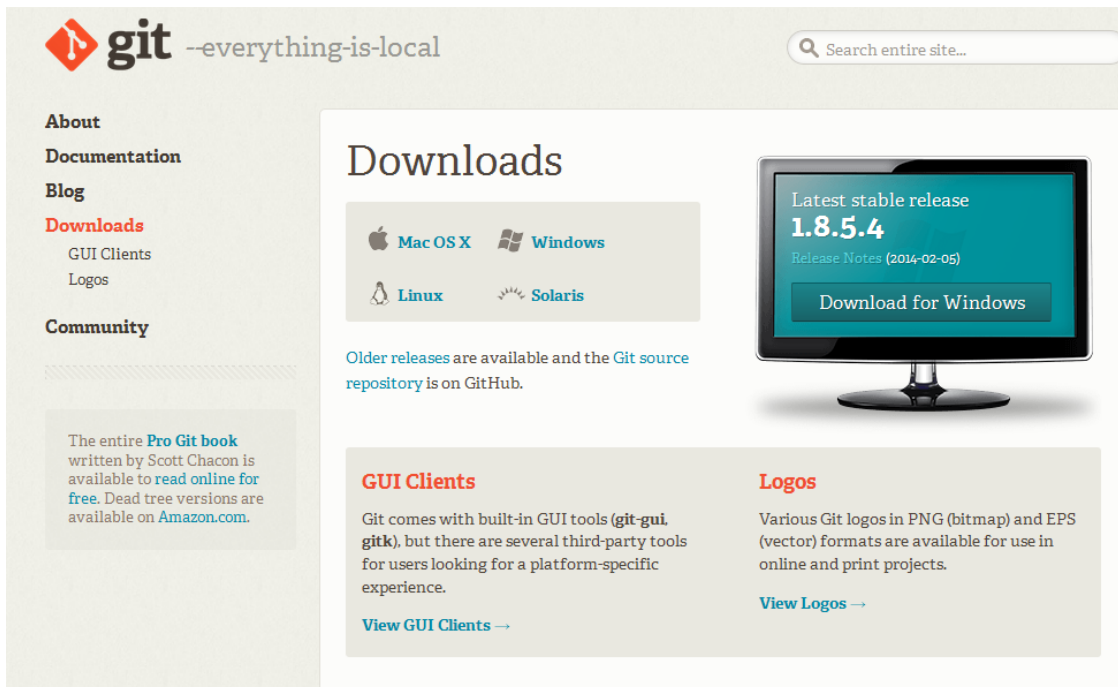
A continuación se nos va a mostrar un formulario para crear nuestro repositorio, escribimos el nombre de como queremos que se llame nuestro repositorio y le damos a create repository, si quieren ponen una descripción.



The screenshot shows the GitHub 'Create new repository' form. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Explore', 'Gist', 'Blog', and 'Help'. On the right of the bar, the username 'deltaodma' is displayed with a plus icon and a copy icon. The form itself has two main sections: 'Owner' and 'Repository name'. The 'Owner' dropdown is set to 'deltaodma'. The 'Repository name' text box contains 'prueba' and has a green checkmark on the right. Below this, a hint says 'Great repository names are short and memorable. Need inspiration? How about **finna-be-bear**.' There is an empty text box for 'Description (optional)'. Below that, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone can see this repository. You choose who can commit.' The 'Private' option is described as 'You choose who can see and commit to this repository.' Below these is a checkbox for 'Initialize this repository with a README', with a note: 'This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.' At the bottom of this section are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by an information icon. At the very bottom is a green 'Create repository' button.

Nos cargara una ventana con los comando para subir nuestro repositorio, antes de esto tendremos que descargar en instalar git, el cual no permitirá subir y actualizar nuestro repositorio.

Seleccionamos la version del sistema operativo desde donde estemos trabajando y descargamo el instalador, la intalacion del git va a tener todos los valores por defecto por lo que solo le vamos a dar siguiente, siguiente sin modificar ningun campo hasta que termine la instalacion.



Una vez instalado git en nuestro equipo vamos y lo abrimos desde el menú inicio, carpeta git, abrimos Git Bash, el cual es una consola desde donde vamos a administrar nuestros repositorios.

Lo primero que aremos en la consola de git será crear la clave ssh, para poder sincronizar nuestro pc con la plataforma github, para esto aremos utilización del comando ssh-keygen.exe.

Nos pedira que le demos un nombre al archivo con nuestra clave ssh publica y una contraseña 2 veces.

```
Welcome to Git (version 1.8.5.2-preview20131230)

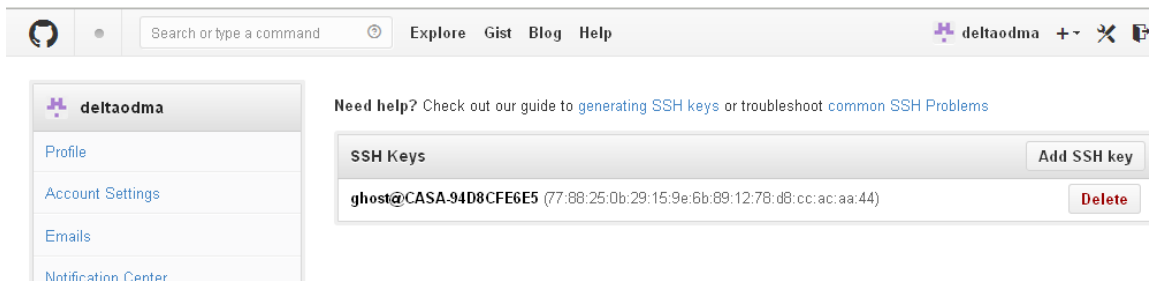
Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

ghost@CASA-94D8CFE6E5 ~
$ ssh-keygen.exe
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Documents and Settings/ghost/.ssh/id_rsa):
Created directory '/c/Documents and Settings/ghost/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Documents and Settings/ghost/.ssh/id_rsa.
Your public key has been saved in /c/Documents and Settings/ghost/.ssh/id_rsa.pub.
The key fingerprint is:
77:88:25:0b:29:15:9e:6b:89:12:78:d8:cc:ac:aa:44 ghost@CASA-94D8CFE6E5
```

Directorio

Después de crear nuestra clave vamos al navegador a la pagina de github al icono de configuración de cuenta ubicado en la parte superior derecha al lado del icono del mas (+) una vez demos clic nos aparecerá un listado con las opciones que podemos editar para modificar nuestra cuenta vamos al apartado de SSH KEYS e introducimos nuestra llave, la cual vamos a encontrar en el directorio que nos muestra la pantalla anterior donde se genera la llave, este directorio varia según nuestro sistema operativo o nombre de usuario. El archivo va a tener el nombre que le hayamos dado o si no le dimos por defecto se va a llamar "id_rsa" tenemos que fijarnos que el tipo de archivo sea ".pub" lo abrimos con un editor de texto y copiamos nuestra calve la cual se vera mas o menos así.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAuj6jhsZl2q+I7fyCsHifFKkstfmIE2REt2hx/sizA5WY7
UQu7k97PioOBiVj/SKohOXRZdqozY9r5z7AzyS7a/+c7/Q9L+gOjoVIGUNpNs3ZwyCvxqeFqt
4DYO3d+iis64shJZqo7aEUagqWNaiNySHAS6GQCngUiL9RWt8lc/Bvlaz/YuopK3zPkvxblK
A5BeOolcy+oxgDkSwehKi4m/ZEvB69hp3JPDJbklGTc4pGz03EI/77gwN9+
```



Agregada nuestra clave nos vamos a configurar git.

```
git config --global user.name "nuestro nombre de usuario(sin comilla)"
git config --global user.email "nuestro email(sin comillas)"
```

Ahora vamos a la carpeta donde se encuentra nuestro proyecto guardado, utilizando los siguientes comandos para desplazarnos por los directorios de nuestro pc

Ls → para que nos muestre una lista de los archivos en un directorio en el que nos encontremos

Cd nombre_del_directorio → para acceder a un directorio

Cd .. → Para retroceder un directorio

Una vez nos encontremos en el directorio que contiene nuestro proyecto iniciamos nuestro repositorio con el comando.

\$ git init

```
$ git init
Initialized empty Git repository in c:/Documents and Settings/ghost/Mis documentos/NetBeansProjects/Vector/.git/
```

Adherimos todos los archivos que vamos a subir en este caso vamos a subir todo por lo que vamos a utilizar el siguiente comando.

\$ git add .

```
ghost@CASA-94D8CFE6E5 ~/Mis documentos/NetBeansProjects/Vector (master)
$ git add .
```

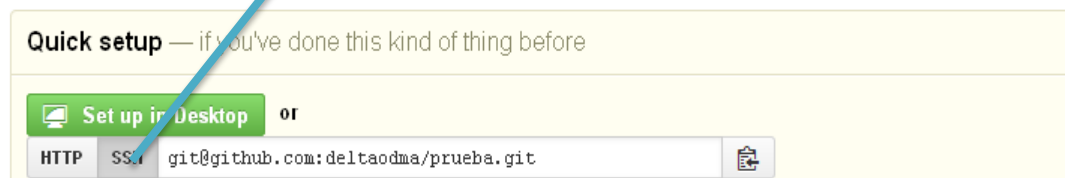
El punto es para que seleccione todos los archivos y carpetas.

Ahora creamos nuestro primer comentario, el comentario sirve como base para saber cuales son los cambios que un usuario le ha hecho al proyecto

\$ git commit -m "nuestro comentario"

```
ghost@CASA-94D8CFE6E5 ~/Mis documentos/NetBeansProjects/Vector (master)
$ git commit -m "Inicio"
[master (root-commit) 680494f] Inicio
13 files changed, 1667 insertions(+)
create mode 100644 build.xml
create mode 100644 build/classes/.netbeans_automatic_build
create mode 100644 build/classes/.netbeans_update_resources
create mode 100644 build/classes/Vetor.rs
create mode 100644 build/classes/vetor/Vetor.class
create mode 100644 manifest.mf
create mode 100644 nbproject/build-impl.xml
create mode 100644 nbproject/genfiles.properties
create mode 100644 nbproject/private/private.properties
create mode 100644 nbproject/private/private.xml
create mode 100644 nbproject/project.properties
```

Ahora le agregamos la dirección ssh de nuestro repositorio la cual nos la dará github



\$ git remote add origin [git@github:deltaodma/prueba.git](https://github.com/deltaodma/prueba.git) → en mi caso.

```
ghost@CASA-94D8CFE6E5 ~/Mis documentos/NetBeansProjects/Vector (master)
$ git remote add origin git@github.com:deltaodma/prueba.git
```

Echo esto ya podemos hacer el push y subir nuestros archivos al repositorio, nos pedirá confirmación y luego nuestra clave de github

\$ git push -u origin master

```
ghost@CASA-94D8CFE6E5 ~/Mis documentos/NetBeansProjects/Vector (master)
$ git push -u origin master
Warning: Permanently added the RSA host key for IP address '192.30.252.129' to the list of known hosts.
Enter passphrase for key '/c/Documents and Settings/ghost/.ssh/id_rsa':
Counting objects: 21, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (21/21), 14.77 KiB | 0 bytes/s, done.
Total 21 (delta 0), reused 0 (delta 0)
To git@github.com:deltaodma/prueba.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Al finalizar el proceso nuestros archivos ya estarán en github y podremos acceder a ellos cuando queramos.

Como se debe de ver nuestro repositorio con los elementos agregados.

The screenshot shows the GitHub interface for a repository named 'prueba' by user 'deltaodma'. At the top, there's a header with the repository name and an 'Unwatch' button. Below this is a section for repository metadata with fields for 'Description' and 'Website', and buttons for 'Save' and 'or cancel'. A summary bar indicates '1 commit', '1 branch', '0 releases', and '1 contributor'. The main content area shows the 'master' branch selected, with a list of files and folders: 'build', 'nbproject', 'src', 'build.xml', and 'manifest.mf'. Each item is listed with its name, the word 'Inicio', and the time '11 minutes ago'. At the bottom, there's a recommendation to 'Add a README' with a corresponding button.

File/Folder	Status	Time
build	Inicio	11 minutes ago
nbproject	Inicio	11 minutes ago
src	Inicio	11 minutes ago
build.xml	Inicio	11 minutes ago
manifest.mf	Inicio	11 minutes ago