

# Cómo crear una cuenta en GitHub, instalar Git, subir un archivo simple y agregar colaboradores.

A continuación se detallan los pasos para crear un repositorio y subir un archivo simple. De este modo, usted debe proceder con sus tareas. El ejemplo está realizado sobre Linux, usando la distribución Ubuntu 12.04 LTS.

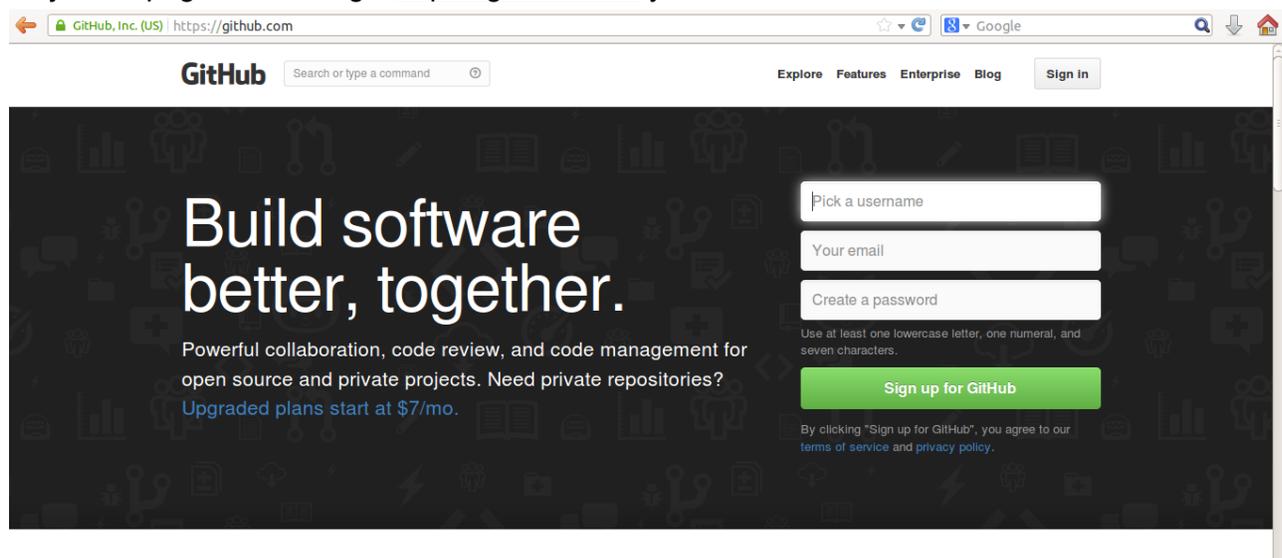
## 1. Introducción

Primero que todo, hay que hacer una aclaración entre las diferencias de **GitHub** y **Git**, **GitHub** es un hosting para Git, que es gratuito para proyectos open source. En cambio, **Git** es un sistema de control de versiones distribuido. Por tanto, esta guía puede ser útil a la hora desarrollar cualquier otro proyecto, ya que el control de versiones es una herramienta sumamente útil a la hora de desarrollar tareas o proyectos que requieran de varios colaboradores.

Normalmente el ramo utiliza el hosting de Assembla para subir las tareas, pero a partir del año 2014 al parecer las políticas de hosting cambiaron y solo se aceptan 3 miembros por proyecto, por lo cual no le da espacio para que su ayudante revise la tarea. Además existen varias empresas que ocupan como hosting a Github.

## 2. Preparar el entorno de Github.

Diríjase al página web de git: <https://github.com/> y cree una cuenta.



The screenshot shows the GitHub website's sign-up page. The browser address bar displays "GitHub, Inc. (US) | https://github.com". The page header includes the GitHub logo, a search bar, and navigation links for "Explore", "Features", "Enterprise", "Blog", and "Sign in". The main content area features the slogan "Build software better, together." and a description: "Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo." The sign-up form consists of three input fields: "Pick a username", "Your email", and "Create a password". Below the password field, there is a note: "Use at least one lowercase letter, one numeral, and seven characters." A green "Sign up for GitHub" button is positioned below the form. At the bottom of the form, a small disclaimer states: "By clicking 'Sign up for GitHub', you agree to our terms of service and privacy policy."

Cada integrante debe crear una cuenta en Github, a su vez tiene que instalar Git, pero solo un integrante debe crear el repositorio (**Paso 4. Creando un Repositorio**).

The screenshot shows the GitHub account creation interface. At the top, there are three progress steps: 'Completed: Set up a personal account', 'Step 2: Choose your plan', and 'Step 3: Go to your dashboard'. The main content is titled 'Choose your personal plan' and features a table of plans. The 'Free' plan is selected and marked as 'Chosen'. To the right, a box lists features included in each plan: unlimited collaborators, unlimited public repositories, free setup, SSL protection, email support, and wikis, issues, pages, and more. Below the table, there is a note: 'Don't worry, you can cancel or upgrade at any time.' and a checkbox for 'Help me set up an organization next' with a link to learn more about organizations. At the bottom, there is a green 'Finish sign up' button.

Plan	Cost	Private repos	
Large	\$50/month	50	<input type="button" value="Choose"/>
Medium	\$22/month	20	<input type="button" value="Choose"/>
Small	\$12/month	10	<input type="button" value="Choose"/>
Micro	\$7/month	5	<input type="button" value="Choose"/>
Free	\$0/month	0	<input type="button" value="Chosen"/>

Each plan includes:

- Unlimited collaborators
- Unlimited public repositories
- Free setup
- SSL Protection
- Email support
- Wikis, Issues, Pages, & more

Don't worry, you can cancel or upgrade at any time.

Help me set up an organization next  
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.  
[Learn more about organizations.](#)

Elija la versión gratuita, haga click sobre el boton de color verde y revise su correo electrónico para confirmar la creación de la cuenta. Por ahora, ya tenemos la cuenta de Github, ahora vamos configurar Git en su computador.

### 3. Preparar el entorno de Git en su computador.

#### 3.1 Instalando Git

Como ubuntu es una distribución basada en Debian, tienes que ejecutar los siguientes comandos desde la terminal:

```
$ sudo apt-get install git
```

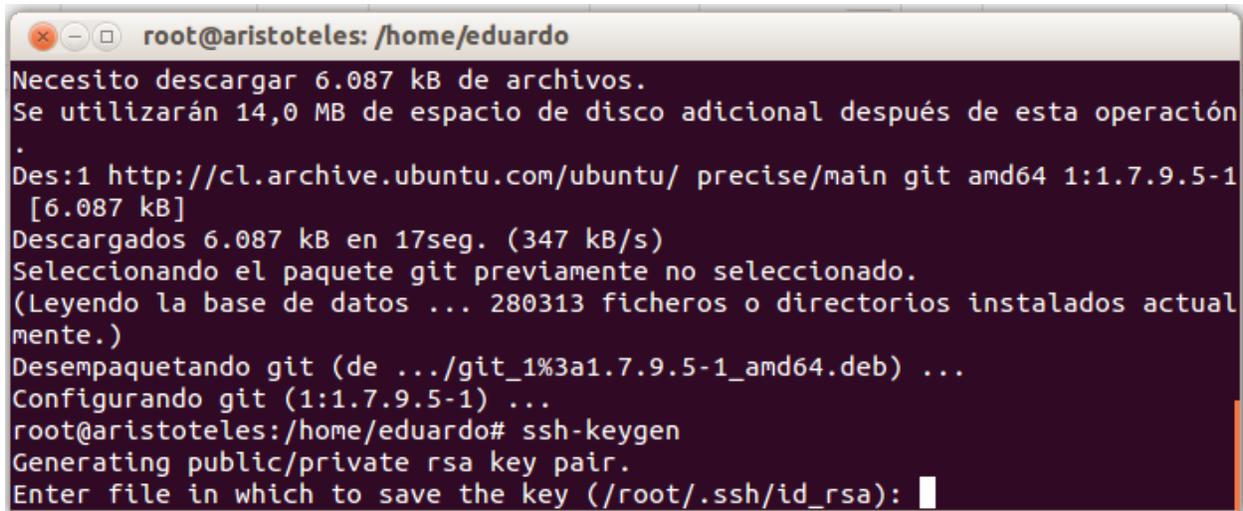
#### 3.2 Creando Llaves

GitHub usa una clave ssh para establecer una conexión segura entre nosotros y el servidor de git. Si no disponemos de una debemos generarla y añadir la misma a nuestro perfil de GitHub.

Para generar una nueva clave, desde el terminal debemos ejecutar los siguientes comandos:

```
$ ssh-keygen
```

al escribir estos comandos, la terminal solicita confirmación en dónde van a guardar las claves (.ssh/id\_rsa):



```
root@aristoteles: /home/eduardo
Necesito descargar 6.087 kB de archivos.
Se utilizarán 14,0 MB de espacio de disco adicional después de esta operación
.
Des:1 http://cl.archive.ubuntu.com/ubuntu/ precise/main git amd64 1:1.7.9.5-1
 [6.087 kB]
Descargados 6.087 kB en 17seg. (347 kB/s)
Seleccionando el paquete git previamente no seleccionado.
(Leyendo la base de datos ... 280313 ficheros o directorios instalados actual
mente.)
Desempaquetando git (de ../git_1%3a1.7.9.5-1_amd64.deb) ...
Configurando git (1:1.7.9.5-1) ...
root@aristoteles:/home/eduardo# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

Luego solicita, dos veces, una contraseña (*passphrase*), contraseña que puedes dejar en blanco si no deseas tener que teclearla cada vez que uses la clave.



```
root@aristoteles: /home/eduardo
d7:ce:8e:88:d5:df:3b:b1:0c:0d:14:05:6f:dc:db:c5 root@aristoteles
The key's randomart image is:
+--[ RSA 2048 ]-----+
|          ..O. |
|         .+. |
|          . E |
|         ..+. |
|        S .O .. |
|       o o. o |
|      . . oo o |
|     o . + .+ |
|    . . . o oo |
+-----+
root@aristoteles:/home/eduardo#
```

### 3.3 Agregando la llave a Github

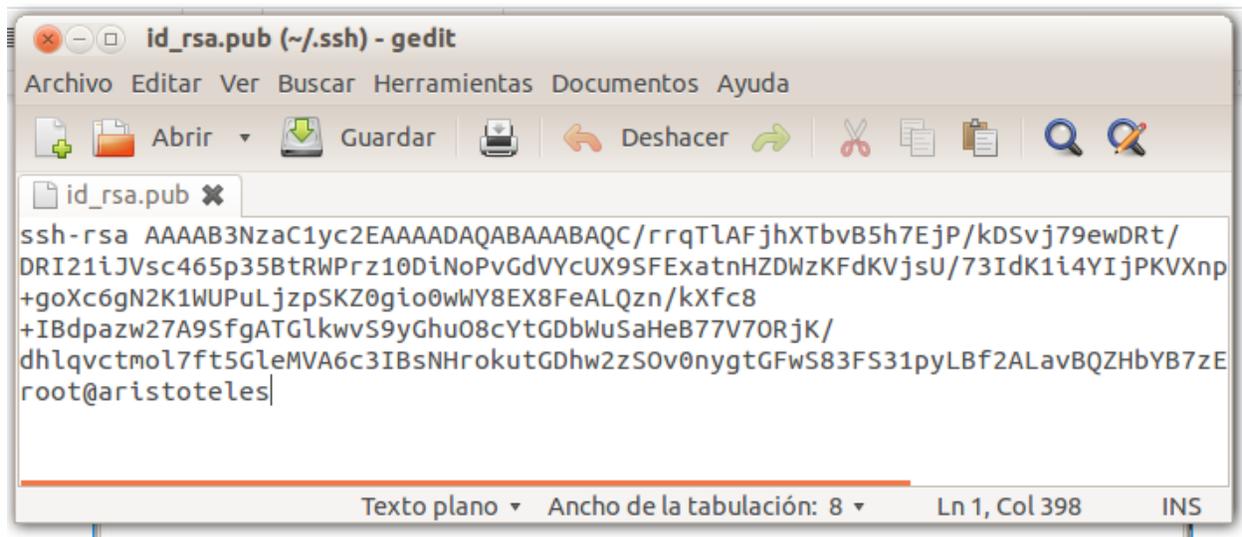
Tras generar la llave, el usuario debe encargarse de enviar su clave pública a quienquiera que administre el servidor Git (en el caso de que éste esté configurado con SSH y así lo requiera). En este caso vamos a copiar la llave al sitio de Github. Para ellos tecleamos los siguiente:

```
$ cd /root/.ssh/
```

Cabe señalar que este comando nos dirige a la carpeta donde se encuentran la llave, si su llave la ubico en otro lugar por favor dirjase a esa carpeta. Ahora que nos encontramos en la carpeta correcta, teclear:

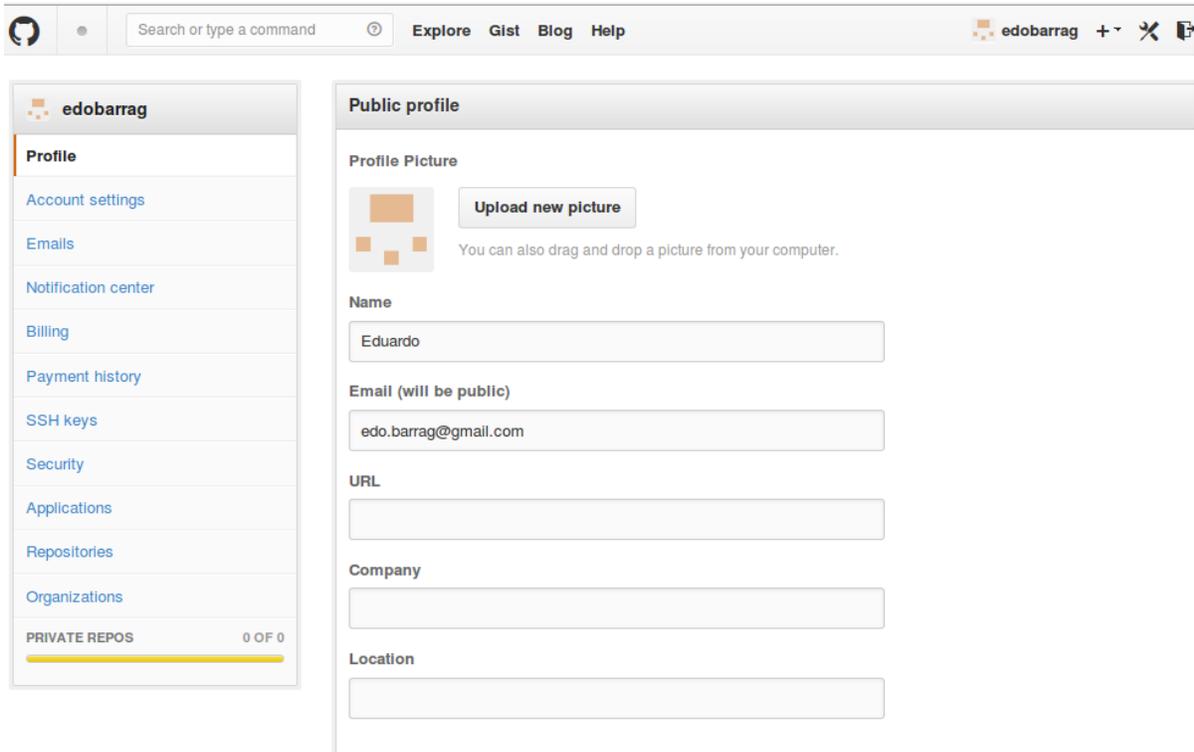
```
$ gedit id_rsa.pub
```

Este comando va a abrir un editor de texto, con la llave que necesitamos copiar. Si todo está resultando bien deberíamos tener algo asi:

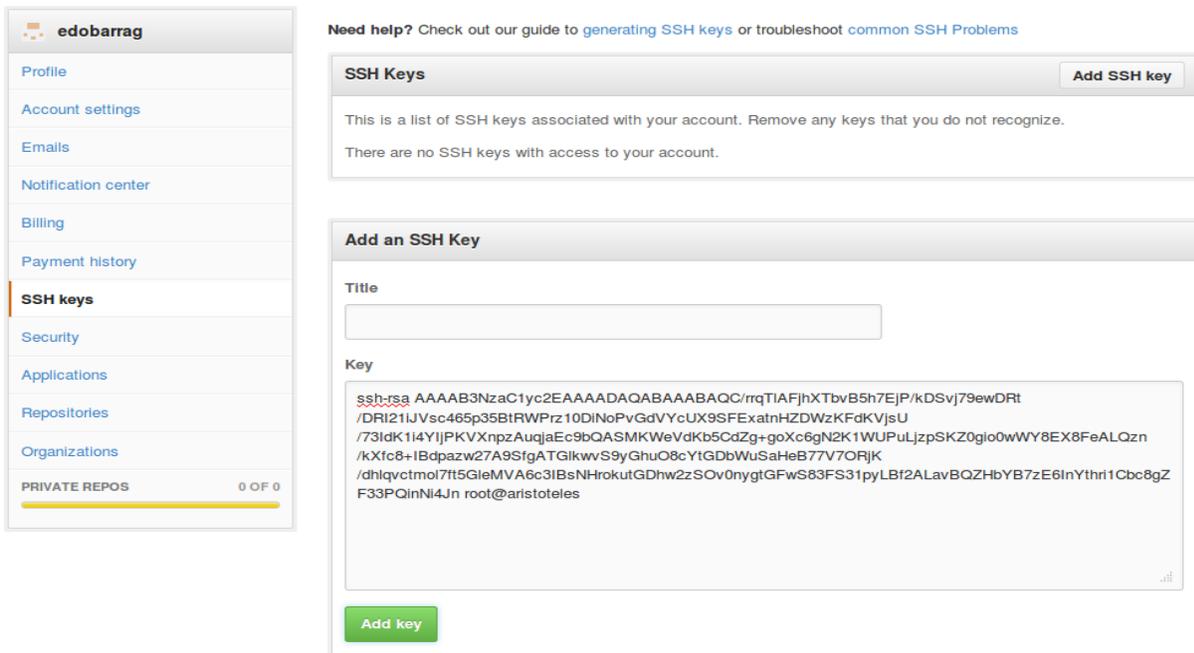


```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC/rrqTlAFjhXTbvB5h7EjP/kDSvj79ewDRt/
DRI21iJVsc465p35BtRWPz10DiNoPvGdVYcUX9SFExatnHZDWzKFdKVjsU/73IdK1i4YIjPKVXnp
+goXc6gN2K1WUPuLjzpSKZ0gio0wwY8EX8FeALQzn/kXfc8
+IBdpazw27A9SfgATGllkwvS9yGhu08cYtGDbWuSaHeB77V70RjK/
dhlqvctmol7ft5GleMVA6c3IBsNHrokutGDhw2zS0v0nygtGFwS83FS31pyLBf2ALavBQZHbYB7zE
root@aristoteles|
```

Ahora vamos a copiar todo el texto comenzando desde las palabras “ssh-rsa ...” hasta la última fila. Una vez listo lo anterior, nos dirigimos al sitio de Github nuevamente y vamos a “Edit your profile” y hacemos click a “SSH keys”:



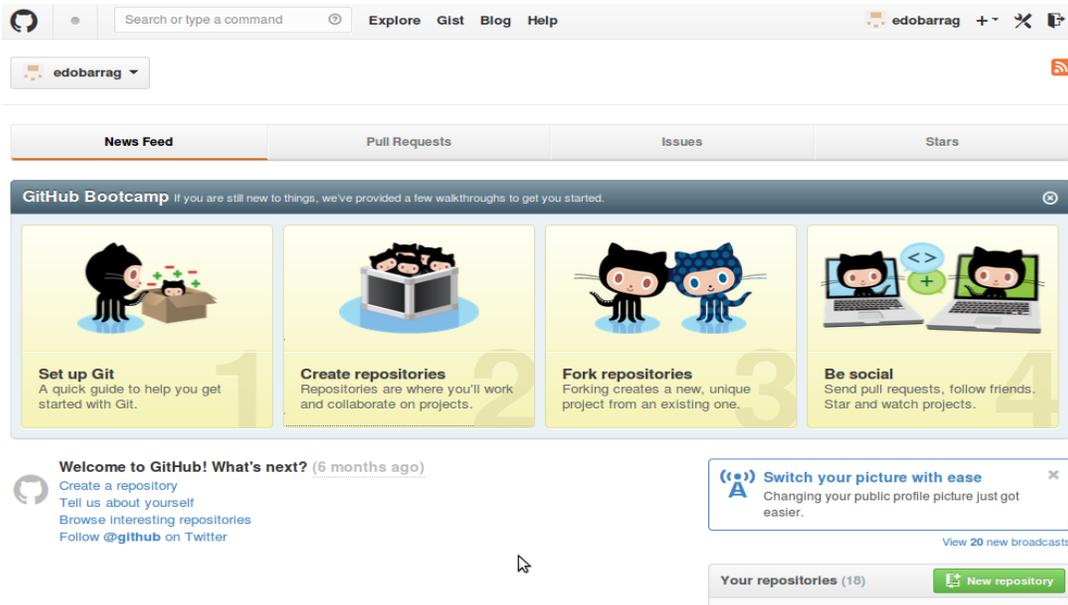
Ahora haga click sobre “Add SSH key” y pegue todo el texto, tal como muestra la figura.



Finalmente haga click sobre “Add key”

#### 4. Creando un Repositorio.

Nos dirigimos al sitio web y creamos un nuevo repositorio, haciendo clic en el botón verde (el botón que está ubicado en la esquina inferior de la imagen).



En Github los repositorios gratuitos son públicos, por ello, ud eventualmente podría clonar cualquier repositorio que sea público. En base a lo anterior y con el fin de evitar que otro grupo se “tiente” a clonar su trabajo, el nombre que le va a asignar a su repositorio tiene que ser lo más aleatorio posible y solo su grupo y el ayudante lo deben conocer.

En este ejemplo, vamos a crear un repositorio llamado G2Tarea1. Hacemos click en “Create repository”, además está la opción de agregar una descripción de la tarea (esto no es necesario).

**Owner** edobarrag / **Repository name** G2Tarea1 ✓

PUBLIC

Great repository names are short and memorable. Need inspiration? How about [glowing-octo-sansa](#).

**Description** (optional)

**Public**  
Anyone can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** | Add a license: **None** ⓘ

**Create repository**

Si todo resulta bien, deberíamos tener algo así :

PUBLIC edobarrag / G2Tarea1

#### Quick setup — if you've done this kind of thing before

or **HTTP** **SSH** `https://github.com/edobarrag/G2Tarea1.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

### Create a new repository on the command line

```
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/edobarrag/G2Tarea1.git
git push -u origin master
```

### Push an existing repository from the command line

```
git remote add origin https://github.com/edobarrag/G2Tarea1.git
git push -u origin master
```

En este momento su repositorio G2Tarea1 está vacío, ahora vamos a agregarle algunos archivos de pruebas. En su computador vamos a crear una carpeta, vaya al directorio y ejecute.

```
$ git init
```

Con ello crearemos el repositorio de git, que no es más que el directorio .git en la misma ubicación (si tuviéramos que volver a crearlo basta con borrarlo).

Ahora vamos a añadir la dirección remota del repositorio de Github. Para ello se deberá indicar dónde estará alojado el repositorio. Escribimos en la consola:

```
$ git remote add origin https://github.com/edobarrag/G2Tarea1.git
```

Ojo con “ edobarrag/G2Tarea1.git ”, que es la dirección de su repositorio, la cuya debería aparecer en la parte superior luego de crear el repositorio, tal como lo señala la imagen.



Una vez escrito los comandos anteriores, podemos comenzar a agregar archivos a la carpeta y luego subirlos.

A modo de ejemplo escriba:

```
$ echo "Este es el primer archivo readme" > README.md  
$ git add README.md  
$ git commit -m "Este es mi primer commit para mi tarea de P00"  
$ git push -u origin master
```

Una vez que escriba estos comandos es posible que le solicite su cuenta de Github. Escriba su username y su clave.

```
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto
d origin https://github.com/edobarrag/G2Tarea1.git
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto# echo "Este es
el primer archivo readme" > README.md
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto# git add READM
E.md
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto# git commit -m
"Este es mi primer commit para mi tarea de POO"
[master (root-commit) 8317fdb] Este es mi primer commit para mi tarea de
POO
1 file changed, 1 insertion(+)
create mode 100644 README.md
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto# git push -u o
rigin master
Username for 'https://github.com': edobarrag
Password for 'https://edobarrag@github.com':
To https://github.com/edobarrag/G2Tarea1.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto#
```

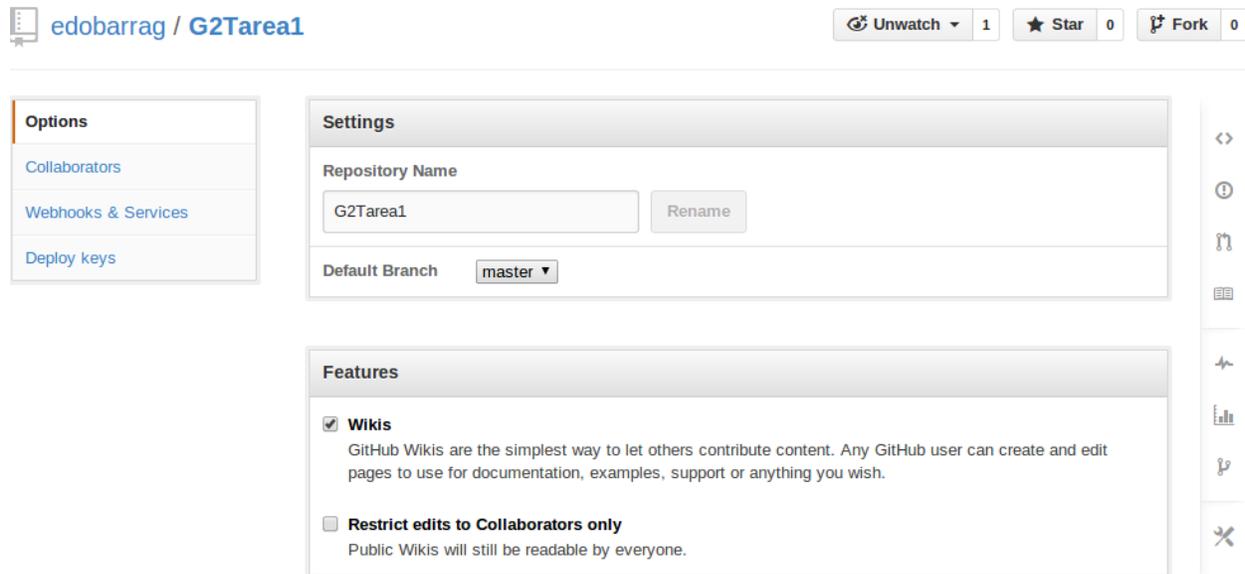
Ahora ya tiene listo su repositorio git con su primer archivo subido (puede revisarlo en la sitio de Github), ahora puede seguir trabajando.

Recuerde que al hacer un "git add" es para indicar a Git cuales serán los archivos que serán incluidos en el siguiente commit (en este caso fue el archivo README.md), luego cuando usted considere que es un buen momento para generar un release, haga un "git commit" y finalmente para hacer la release accesible desde el repositorio remoto haga un "git push".

## 5. Agregando a su compañero para participar en el mismo repositorio.

### 5. 1 Obtener acceso como colaborador

El dueño del repositorio en el que vamos debe dar acceso a sus compañero colaboradores. Para ello deberá acceder al repositorio, entrar en settings/collaborators y añadirnos -add a friend- a la lista.



Le solicitará una contraseña, luego de ello agregue el username de su compañero de trabajo. El colaborador recibirá una notificación. Tiene que agregar tanto a sus compañeros como a su ayudante. El username del ayudante es: **edobarrag**

Lo que resta de la guía está enfocada al colaborador.

### 5. 1 Clonar Repositorio y hacer aportes.

Uds recibirá una notificación de que ahora puede hacer aportes al repositorio. En la carpeta que usará para trabajar, deberá escribir en consola:

```
$ git init
$ git clone https://github.com/edobarrag/G2Tarea1.git
```

Esto permite iniciar Git y clonar el repositorio en su computador. Por favor diríjase a la carpeta correspondiente y modifique el archivo README.md

```
$ gedit README.md
$ git add README.md
$ git commit -m "Este es mi segundo commit para mi tarea de P00"
$ git push -u origin master
```

Si todo resulta bien la consola debería pedir que escriba su username y clave de Github, tal como lo muestra la siguiente imagen:

```
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto/G2Tarea1
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto/G2Tarea1# gedit README.md
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto/G2Tarea1# git add README.md
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto/G2Tarea1# git commit -m "Este es mi segundo commit para mi tarea de P00"
[master ac07ea1] Este es mi segundo commit para mi tarea de P00
 1 file changed, 1 insertion(+), 1 deletion(-)
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto/G2Tarea1# git push -u origin master
Username for 'https://github.com': xhu4rg0
Password for 'https://xhu4rg0@github.com':
To https://github.com/edobarrag/G2Tarea1.git
 07d2983..ac07ea1  master -> master
Branch master set up to track remote branch master from origin.
root@aristoteles: /home/eduardo/Descargas/Pruebas/proyecto/G2Tarea1#
```

Ahora puede revisar en el sitio web el commit realizado. Por tanto, ya puede realizar su tarea de POO y desarrollar su código, tal como se hace hoy en día en numerosas empresas. No olvide **mantener en secreto el nombre de su repositorio y enviar este nombre al ayudante.**

Finalmente, existen un sin numero de cosas entretenidas que Ud puede realizar con Git por ello, lo insto a que revise el material que abunda en la web acerca de Git.

Un link de interes es:

<http://geneura.ugr.es/~jmerelo/tutoriales/git/>

Si por casualidad encuentra algún error en esta guía, comuníquese con el ayudante para que la corrija.