

Envío de proyectos a GitHub desde la terminal mediante protocolo SSH

Funcionamiento del protocolo SSH

SSH es un protocolo de transmisión de datos que utiliza un sistema de doble clave; clave pública y privada.

- Clave pública: Se almacena en tu dispositivo y en tu cuenta de GitHub
- Clave privada: Sólo se almacena en tu dispositivo

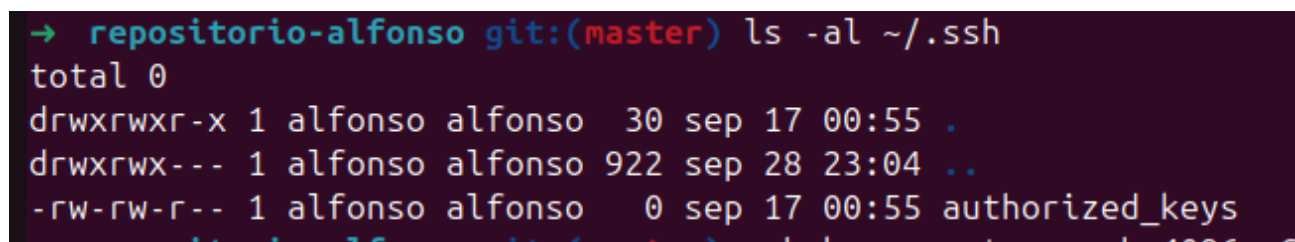
Procedimiento de autenticación

1. Mi ordenador establece una petición de conexión SSH con GitHub.
2. GitHub recibe la petición y genera un “desafío”. Este desafío es un mensaje cifrado, que se cifra con la ayuda de la clave pública.
3. GitHub manda el desafío a mi ordenador.
4. Para descifrar el desafío es necesario tener la clave privada, por eso mi ordenador es capaz de descifrarlo.
5. Tu ordenador envía una respuesta a GitHub de que realmente a ha logrado descifrar el desafío.
6. Se establece la conexión SSH entre GitHub y el ordenador.
7. Una vez establecida la conexión se permite el envío de archivos entre GitHub y tu ordenador. Estos archivos se mandan de forma cifrada y es el receptor (GitHub o tu ordenador) quien se encarga de descifrar la información.

Establecimiento de la conexión SSH

A) Comprobar la presencia de claves públicas y privadas almacenadas: Ejecutar comando:

`ls -al ~/.ssh`



```
→ repositorio-alfonso git:(master) ls -al ~/.ssh
total 0
drwxrwxr-x 1 alfonso alfonso 30 sep 17 00:55 .
drwxrwx--- 1 alfonso alfonso 922 sep 28 23:04 ..
-rw-rw-r-- 1 alfonso alfonso  0 sep 17 00:55 authorized_keys
→ repositorio-alfonso git:(master) ssh-keygen -t rsa -b 4096 -C
```

B) Creación de la clave pública y privada:

Se trata de generar una clave pública y otra privada (asociadas ambas a una cuenta GitHub mediante el e:mail), y almacenarlas en el ordenador. Para ello ejecutar comando:

```
ssh-keygen -t rsa -b 4096 -C "manzano.rodriquez.alfonso.ismael.iescamas.es"
```

```
→ repositorio-alfonso git:(master) ssh-keygen -t rsa -b 4096 -C "manzano.rodriquez.alfonso.ismael.iescamas.es"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alfonso/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alfonso/.ssh/id_rsa
Your public key has been saved in /home/alfonso/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:mXVnB5ot51kw0XSZDLbdwxx7AuBXmRsnT+U2lK4k01M manzano.rodriquez.alfonso.ismael.iescamas.es
The key's randomart image is:
+---[RSA 4096]-----+
|      .+O*B|
|      . *&@*|
|      o *.B/B|
|      + + E.B=|
|      S  = + |
|      + . |
|      o |
|      |
+-----[SHA256]-----+
```

Posteriormente hacemos click en “enter” dos veces, la primera para aceptar el directorio donde se almacenan dichas claves y la segunda para establecer si queremos un mensaje de confirmación.

C) Iniciar el agente SSH:

Este proceso es un programa que permite que no tengamos que introducir mensajes de confirmación cada vez que queremos establecer una conexión SSH. Para iniciar este proceso ejecutamos el comando:

```
eval "$(ssh-agent -s)"
```

```
→ repositorio-alfonso git:(master) eval "$(ssh-agent -s)"
Agent pid 20605
```

D) Añadir la clave privada al agente SSH:

Una vez iniciado el proceso (agente SSH), es necesario cargar la clave privada sobre él. Para ello ejecutamos el siguiente comando:

```
ssh-add ~/.ssh/id_rsa
```

```
→ repositorio-alfonso git:(master) ssh-add ~/.ssh/id_rsa
Identity added: /home/alfonso/.ssh/id_rsa (manzano.rodriquez.alfonso.ismael.iescamas.es)
```

E) Copiar la clave pública:

Mostraremos en la terminal el contenido de la clave pública mediante el comando:

```
cat ~/.ssh/id_rsa.pub
```

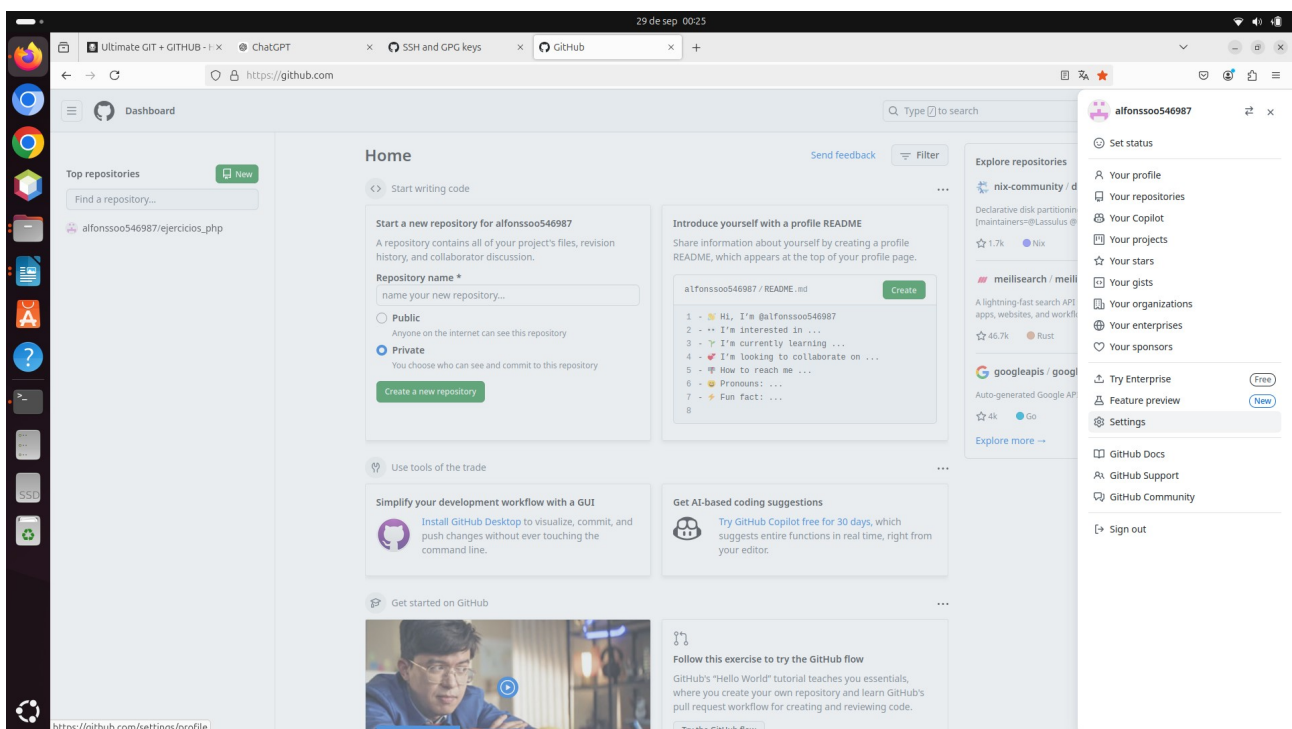
```

→ repositorio-alfonso git:(master) cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQDLXt9E+airjE9ZVlRkQs8UP5oVvo9LUZdfQhdHh8kl/C9/coju9xsE8PzXOGmbzYxHMDrgrq88XvTnn6osFvQ7hsJUHvYx6yI8SKN1/1pP1F7N9pBeqkj0TzdQaRR39QgExRu4xs1Pu6P
vuZs03yCOVpS8p0DxeltjGtIwoxt//Zym/KqxXQ/ut0RZVvFmXnSUmvti9sYQr96tMn2Cnd0J5pa0ybVjrgtHl/3jvyZILRCrLkV69Pwtpv5N17Zs1SsC40sZ6J3xMLTQskesnwlp19NeApoF162cf1aJz5m0eND9R1kd9tKx1T4MF7JQmosADV
crQckVwOUDYxtr0hVf0ce6199U7dZo70jAbA41av6jUw8BmPlqHl27sS07cRrLxSNl1jHfwpDvayDDQu3U400yWZT4a1pfY6UpX1bwCL9qM5QoogFCCPZyGpm2f41Fh003tAmPedvp2UitRGraL9LHtB6gpg8Yqq3L+Vlaa/S0VU6PzH9jN6/imK
1uKXn6+YZ3Hg/dHMY1xEn/SVeZ3ab3baeqknjC9tEuqxNuo4uphIe10f6TqJ3fjFZdV4q56Kf4T1F+Cqt10KexjuiqXEfqTD1Rw0z3XZNVKkz++FePfKjoy8q2UbVVRNtpes1txa/2nBS+Ruz0TH7Ban4bSa/FEcxzdUw== manzano.rod
riquez.alfonso.ismael.iescamas.es

```

Posteriormente seleccionamos el contenido que devuelve la terminal y hacemos un “copy”.

- Almacenar la clave pública en la cuenta de GitHub: Vamos a pegar la clave pública copiada en la terminal en nuestra cuenta GitHub, para ello seguimos los siguientes pasos:
- Ir e iniciar nuestra cuenta GitHub (<https://github.com/>).
- Ir a nuestro perfil → Settings → SSH and GPG keys (a la izquierda) → New SSH keys
- Añadir un título inventado (Title) y pegar en la caja (Key) el contenido previamente copiado del comando “cat ~/.ssh/id_rsa.pub”.
- Add SSH key



29 de sep 00:27

Ultimate GIT + GITHUB - 1 x ChatGPT SSH and GPG keys Your profile

https://github.com/settings/profile

Settings

alfonsoo546987 (alfonsoo546987)
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security

Public profile

Name

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

Select a verified email to display

You can manage verified email addresses in your [email settings](#).

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

Pronouns

Don't specify

URL

ORCID iD

ORCID provides a persistent identifier - an ORCID iD - that distinguishes you from other researchers. Learn more at [ORCID.org](#).

Connect your ORCID iD

Social accounts

Link to social profile

Link to social profile

Link to social profile

Profile picture

Edit

https://github.com/settings/keys

29 de sep 00:28

Ultimate GIT + GITHUB - 1 x ChatGPT SSH and GPG keys SSH and GPG keys

https://github.com/settings/keys

Settings

alfonsoo546987 (alfonsoo546987)
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

Laptop Ubuntu

SHA256: eXVnB5otS1kwXSZDLbdwxx7AuBxRnT+U21K4k03M

Added on Sep 28, 2024

Never used — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

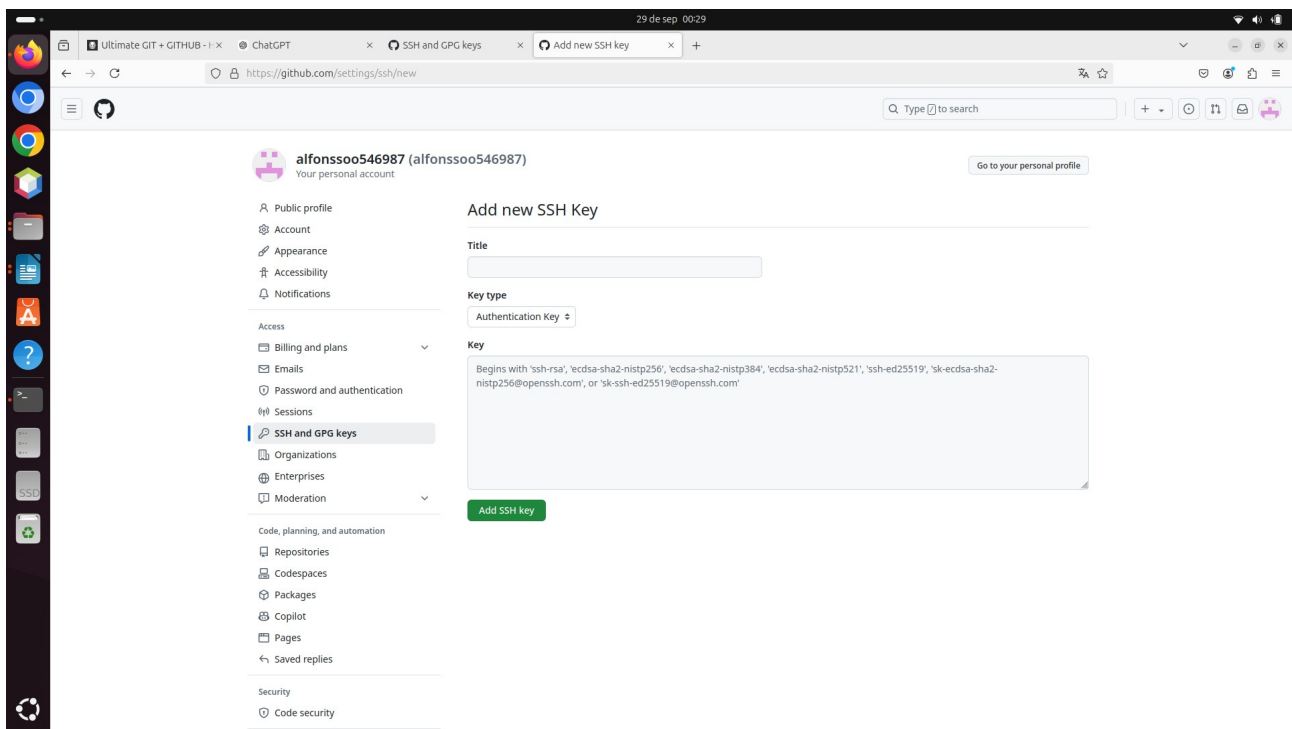
Learn how to [generate a GPG key](#) and [add it to your account](#).

Vigilant mode

☐ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits. [Learn about vigilant mode](#).

https://github.com/settings/ssh/new



F) Probar la conexión

Intentaremos establecer la conexión SSH con la cuenta GitHub mediante el comando:

`ssh -T git@github.com`

Si todo es correcto aparecerá un mensaje tal y como “Hi alfonso546987! You've successfully authenticated, but GitHub does not provide shell access.”

```
→ repositorio-alfonso git:(master) ssh -T git@github.com
Hi alfonso546987! You've successfully authenticated, but GitHub does not provide shell access.
```

también podemos comprobar si las claves fueron almacenadas introduciendo de nuevo el comando :

`ls -al ~/.ssh`

```
→ repositorio-alfonso git:(master) ls -al ~/.ssh

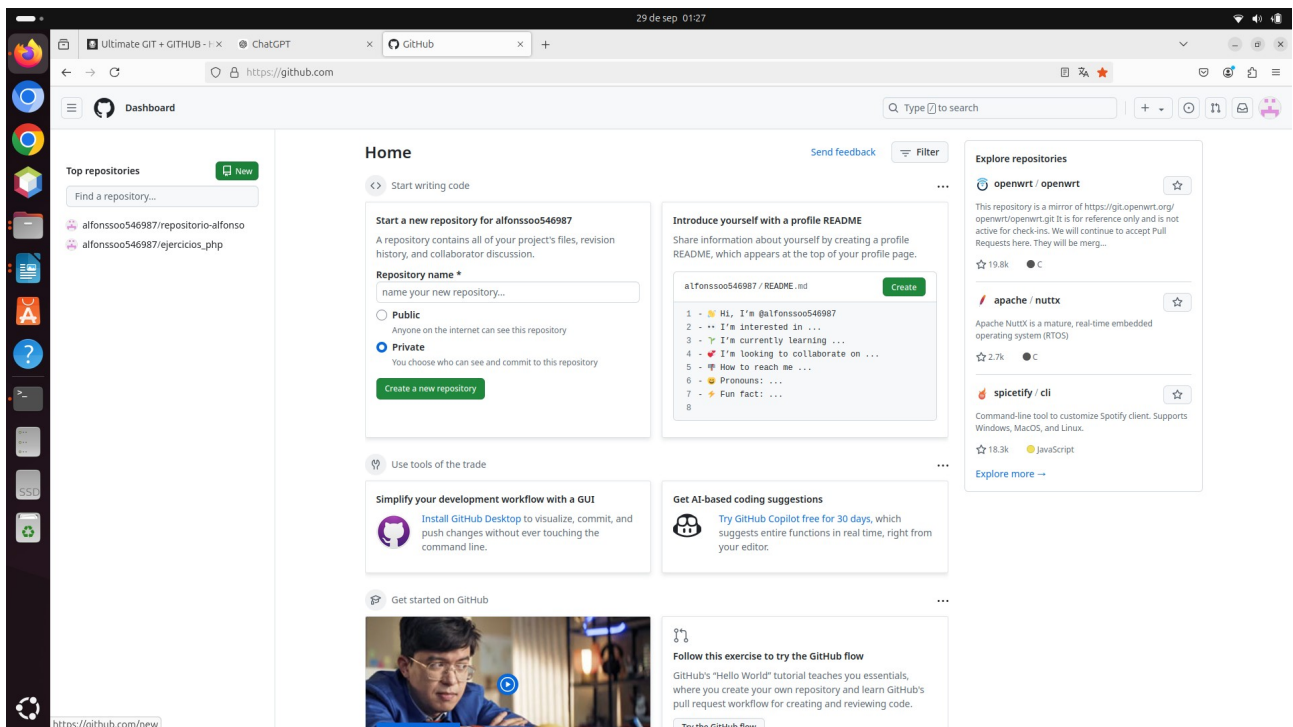
total 16
drwxrwxr-x 1 alfonso alfonso 114 sep 28 23:24 .
drwxrwx--- 1 alfonso alfonso 922 sep 28 23:54 ..
-rw-rw-r-- 1 alfonso alfonso  0 sep 17 00:55 authorized_keys
-rw----- 1 alfonso alfonso 3422 sep 28 23:08 id_rsa
-rw-r--r-- 1 alfonso alfonso 770 sep 28 23:08 id_rsa.pub
-rw----- 1 alfonso alfonso 978 sep 28 23:24 known_hosts
-rw-r--r-- 1 alfonso alfonso 142 sep 28 23:17 known_hosts.old
```

Subir los proyectos a GitHub

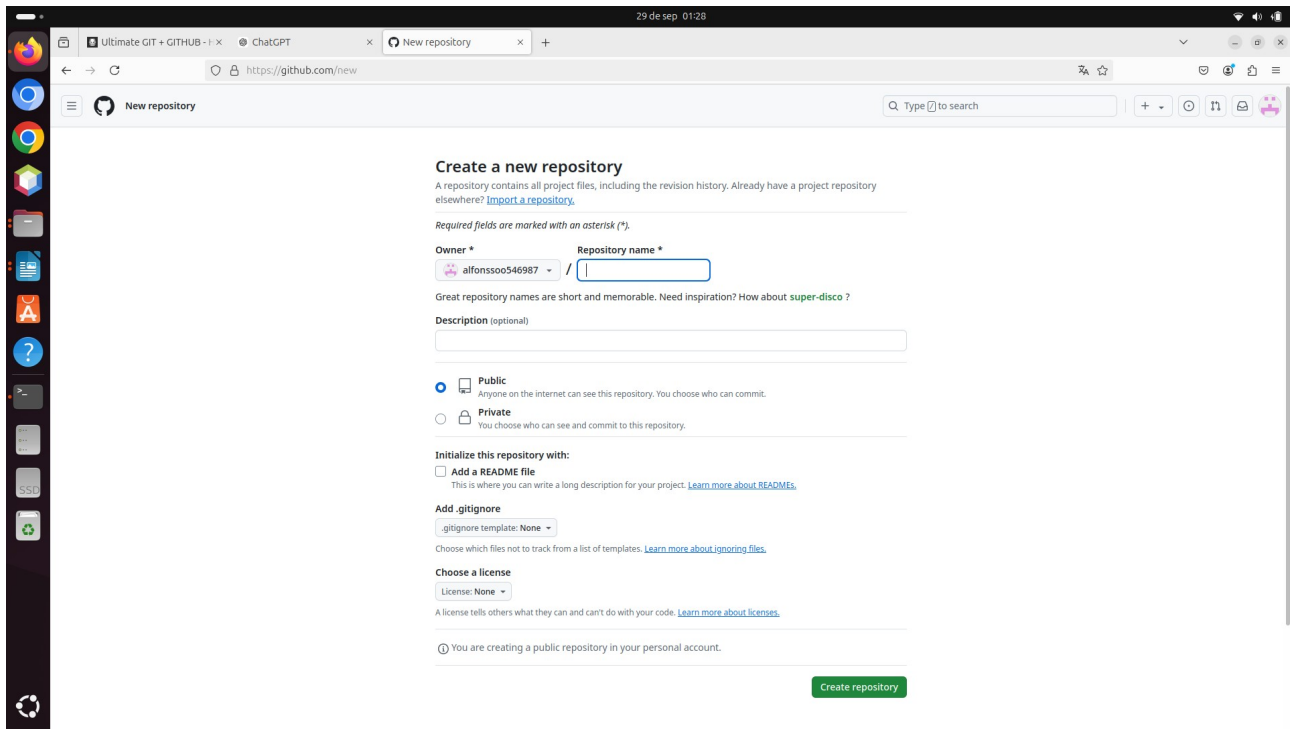
Una vez hemosestablecido una conexión SSH los archivos se transfirirán entre tu ordenador y GitHub mediante este protocolo, no hay que hacer nada más.

Subir un commit de un repositorio local (desde tu ordenador) a un repositorio de repositorio de GitHub se denomina “hacer un push”. Cuando haces un push subes el estado de la rama que definas.

Para poder subir los cambios que vas haciendo en tu repositorio local, primero debes crear un repositorio en GitHub donde subirlo. Este repositorio que creas en GitHub debe tener el mismo nombre que tu repositorio local. Para añadir el repositorio ve a tu cuenta GitHub y click en “New repository”.



Dale un nombre al repositorio y crealo.



Ya podras subir contenido mediante el comando:

`git push X Y`

X: Es el nombre predeterminado que Git asigna al repositorio remoto original que se ha clonado o añadido. Puede ser renombrado, pero "origin" es el nombre más común. Para conocer el nombre utiliza este comando :

`git remote -v`

```
→ repositorio-alfonso git:(master) git remote -v
origin  git@github.com:alfonso546987/repositorio-alfonso.git (fetch)
origin  git@github.com:alfonso546987/repositorio-alfonso.git (push)
```

Aquí vemos que X = “origin”.

Y: Es el nombre de la rama que quieres subir. La rama principal se llama “main” o “master”. Para saber cual de las dos usar ejecuta este comando:

`git branch`

Se abrirá un archivo con el nombre de la rama. En este caso Y = “master”.

Por ello para poder hacer el “push” ejecutaremos el siguiente comando:

`git push origin master`

Para comprobar si el contenido ha sido subido a GitHub haremos click en el repositorio creado.

Hacer un fork

Un fork consiste en buscar un repositorio remoto de algún autor que te interese y guardar una copia del mismo en tu listado de repositorios GitHub. Los cambios que realice el autor pasarán a tu cuenta.

En la barra superior puedes buscar autores y acceder a sus cuentas.

Busca sus repositorios y haz click en la pestaña “fork” para que este se copie en tu listado de repositorios.

Opcionalmente puedes sincronizar el repositorio forkeado en tu equipo local. Para ello usarás el comando “git clone”.