



CIPFP Mislata

Centre Integrat Públic
Formació Professional Superior

Tutorial Git

Autor: Joan Puigcerver Ibáñez

Correo electrónico: j.puigcerveribanez@edu.gva.es

Curso: 2022/2023

Licencia: BY-NC-SA

(Reconocimiento - No Comercial - Compartir Igual)



1.Objetivos

El objetivo de este ejercicio es entender los conceptos básicos para trabajar en Git, así como crear nuestro primer repositorio y configurar el acceso y la autenticación.

2.Instalación de Git

Para poder utilizar Git, primero, es necesario instalarlo.

2.1. Windows

Puede descargarlo de la página oficial: <https://git-scm.com/>

Una vez instalado, puede utilizar la herramienta **Git Bash** para realizar el tutorial (funciona al igual que una terminal Linux).

2.2. Ubuntu (Debian)

Puede descargarlo e instalarlo mediante `apt`.

```
sudo apt update  
sudo apt install git
```

Una vez instalado, el mandato `git` debería estar disponible.

3. GitHub

[GitHub](#) es un servicio de hosting de repositorios Git, el cual ofrece toda la funcionalidad de Git de control de revisión distribuido y administración de código de la fuente (SCM) así como añadiendo sus características propias. A diferencia de Git, que es estrictamente una herramienta de línea de comandos, GitHub proporciona una interfaz gráfica basada en web y escritorio, así como integración del móvil.

3.1. Usuario

Ante todo, es necesario crear (si no tiene ya) un usuario en la plataforma GitHub.

- Realiza la tarea **Usuario de GitHub** e indica a tu usuario.

3.2. Tutorial GitHub

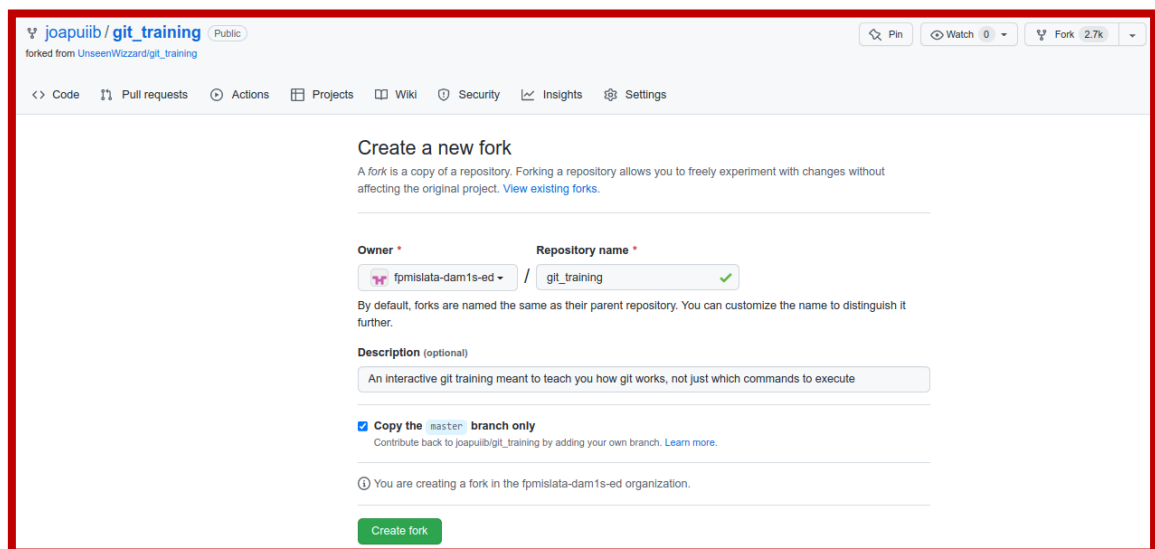
Este ejercicio consiste en realizar el tutorial de GitHub, que puede encontrar:

- **Versión original (en inglés):** https://github.com/joapuiib/git_training
- **Versión traducida (en valenciano):** https://github.com/joapuiib/git_training/tree/master/translations/ca

Le recomiendo que trabaje con la versión en inglés.

Para poder trabajar desde su usuario, debe realizar un **fork** en su cuenta.

- [Cómo hacer un fork en GitHub](#)



En la captura se utiliza la organización **fpmislata-dam1s-ed** como propietario para realizar el ejemplo. En su caso, debe seleccionar su usuario.

Antes de realizar el tutorial, es necesario configurar la autenticación de alguna de las maneras que se exponen a continuación.

Le recomiendo utilizar el método SSH, que es más difícil de configurar, pero más seguro y funciona perfectamente en el centro.

3.3. Autenticación GitHub

Para poder *clonar* repositorios privados o para poder realizar cambios (*push*) en cualquier repositorio, debe autenticarse.

Desde el 13 de agosto de 2021, GitHub ha eliminado la posibilidad de identificarse mediante usuario y contraseña. Consulte [Support for password authentication was removed. Please use personal access token instead.](#)

Por tanto, podemos utilizar dos mecanismos para identificarnos.

3.3.1. SSH

Documentación oficial: [Conectar a GitHub con SSH](#)

Generar claves SSH

En **Linux** o **Mac OS**, debemos utilizar la herramienta `ssh-keygen` que puede ejecutar en la línea de pedidos del sistema.

```
$ ssh-keygen -t rsa -b 4096
```

Le pedirá introducir un nombre personalizado o utilizar una contraseña para desbloquear esta clave. Todas estas opciones son configurables y no son obligatorias.

En el ejemplo, crearemos una clave llamada **ssh_ed**, sin contraseña.

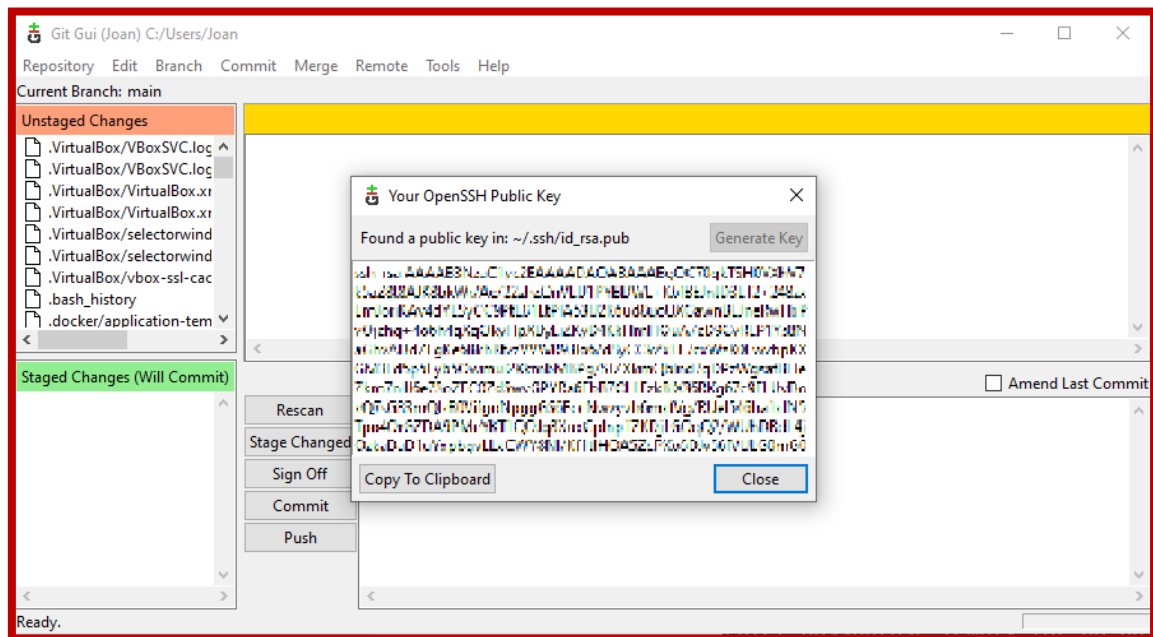
```
jpuigcerver@0A6PC02:~/.ssh$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jpuigcerver/.ssh/id_rsa): damls_ed
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in damls_ed
Your public key has been saved in damls_ed.pub
The key fingerprint is:
SHA256:TtVjgWe9XDtBmq1XPT+2UncS9GZrvio/Lqh5XK7Hoa4 jpuigcerver@0A6PC02.ad.fpmislata.com
The key's randomart image is:
+---[RSA 4096]---+
|           ...0. |
|          ..00=00|
|         .o++.=0 |
|        . . .+*= |
|       S  ..B=  |
|      o o  *. = |
|     o *  .. o  |
|    . = *  .. . |
|   E=00  =+o.  |
+-----[SHA256]-----+
```

Las claves predeterminadas se generan en el directorio **.ssh**. En caso de que se quiera personalizar el nombre de la clave, es necesario introducir la ruta donde se quiere guardar. En el ejemplo, se han creado desde dentro del directorio **.ssh** para que se generen dentro del mismo.

Verá que se han generado dos archivos:

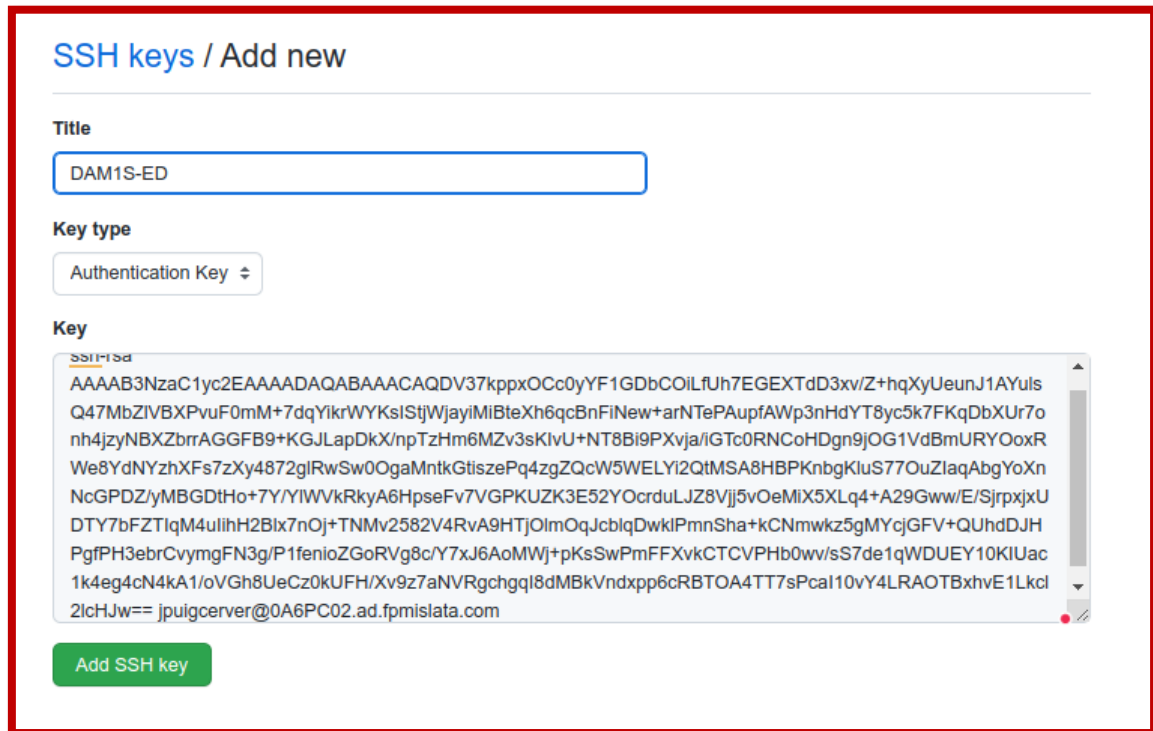
- **dam1s_ed**: Clave privada, este archivo no se debe compartir con nadie ya que es el que se utiliza para identificarse.
- **dam1s_ed.pub**: Clave pública, que se utiliza con la clave privada para realizar la autenticación. Esta clave debe proporcionarse a los equipos/servicios donde te quieres autenticar (en este caso en GitHub).

En cambio, en **Windows** puede generar la clave mediante **Git GUI**. Es necesario acceder a **Help > Show SSH Key**, que nos abrirá una ventana donde podemos generar una clave. Una vez generada la clave, nos mostrará el contenido de la **clave pública** que se va a utilizar en el siguiente apartado.



Introducir clave SSH en GitHub

Para configurar una clave SSH en Github es necesario ir a **Settings => SSH and GPG keys => New SSH key**. En esta página, es necesario indicar un nombre, el tipo (Authentication Key) e introducir el contenido de la **clave pública** generada.



SSH keys / Add new

Title

DAM1S-ED

Key type

Authentication Key

Key

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADV37kppxOCc0yYF1GDbCOiLfUh7EGEXTdD3xv/Z+hqXyUeunJ1AYuls
Q47MbZlVBXPvuF0mM+7dqYikrWYKsISTjWjayiMiBteXh6qcBnFiNew+arNtePAupfAWp3nHdYT8yc5k7FK
qDbXUr7onh4jzyNBXZbrrAGGFB9+KGJLapDkX/npTzHm6MZv3sKivU+NT8Bi9PXvja/iGTc0RNCohDgn9jOG1VdBmURYOoxR
We8YdNYzhXF7zXy4872glRwSw0OgaMntkGtiszePq4zgZQcW5WELyI2QtMSA8HBPknbgKluS77OuZlaqAbgYoXn
NcGPDZ/yMBGDtHo+7Y/YlWVvRkyA6HpseFv7VGPKUZK3E52Y0crduLJZ8Vjj5v0eMiX5XLq4+A29Gww/E/SjrpjxU
DTY7bFZTIqM4ulihH2Blx7nOj+TNMv2582V4RvA9HTjOlmOqJcblqDwklPmnSha+kCNmwkz5gMYcjGFV+QUhdDJH
PgFPH3ebrCvymgFN3g/P1fenioZGoRVg8c/Y7xJ6AoMWj+pKsSwPmFFXvkCTCVPHb0wv/sS7de1qWDUEY10KIUac
1k4eg4cN4kA1/oVGh8UeCz0kUFH/Xv9z7aNVrgchggI8dMBkVndxpp6cRBT0A4TT7sPcal10vY4LRAOTBxhvE1Lkcl
2lcHJw== jpuigcerver@0A6PC02.ad.fpmislata.com

Add SSH key

Una vez configurada la clave, podemos clonar repositorios mediante SSH. Fíjese con que la URL no sigue el mismo formato que con el protocolo HTTPS.

```
jpuigcerver@0A6PC02:~/.ssh$ cat dam1s.ed.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADV37kppxOCc0yYF1GDbCOiLfUh7EGEXTdD3xv/Z+hqXyUeunJ1AYuls
Q47MbZlVBXPvuF0mM+7dqYikrWYKsISTjWjayiMiBteXh6qcBnFiNew+arNtePAupfAWp3nHdYT8yc5k7FK
qDbXUr7onh4jzyNBXZbrrAGGFB9+KGJLapDkX/npTzHm6MZv3sKivU+NT8Bi9PXvja/iGTc0RNCohDgn9jOG1VdB
mURYOoxRWe8YdNYzhXF7zXy4872glRwSw0OgaMntkGtiszePq4zgZQcW5WELyI2QtMSA8HBPknbgKluS77OuZla
qAbgYoXnNcGPDZ/yMBGDtHo+7Y/YlWVvRkyA6HpseFv7VGPKUZK3E52Y0crduLJZ8Vjj5v0eMiX5XLq4+A29Gww/
E/SjrpjxUDTY7bFZTIqM4ulihH2Blx7nOj+TNMv2582V4RvA9HTjOlmOqJcblqDwklPmnSha+kCNmwkz5gMYcjG
FV+QUhdDJHPgFPH3ebrCvymgFN3g/P1fenioZGoRVg8c/Y7xJ6AoMWj+pKsSwPmFFXvkCTCVPHb0wv/sS7de1qWD
UEY10KIUac1k4eg4cN4kA1/oVGh8UeCz0kUFH/Xv9z7aNVrgchggI8dMBkVndxpp6cRBT0A4TT7sPcal10vY4LRA
OTBxhvE1Lkcl2lcHJw== jpuigcerver@0A6PC02.ad.fpmislata.com
jpuigcerver@0A6PC02:~/.ssh$ cd ..
jpuigcerver@0A6PC02:~$ git clone git@github.com:joapuiib/git_training.git
Cloning into 'git_training'...
remote: Enumerating objects: 240, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 240 (delta 13), reused 35 (delta 11), pack-reused 202
Receiving objects: 100% (240/240), 2.11 MiB | 586.00 KiB/s, done.
Resolving deltas: 100% (68/68), done.
jpuigcerver@0A6PC02:~$ cd git_training/
jpuigcerver@0A6PC02:~/git_training$ ls
Alice.txt  img  README.md  translations
jpuigcerver@0A6PC02:~/git_training$
```

3.3.2. Personal Acceso Token (PAT)

La primera respuesta de [Support for password authentication was removed. Please use personal access token instead.](#) explica muy bien cómo puede crear un PAT y cómo configurarlo en su sistema.

Para crear un PAT debe ir a **Settings => Developer Settings => Personal Access Token => Generar New Token (Give your password)**. En esta página, llenar el formulario (es necesario seleccionar las opciones *repo*) y darle a **Generar token**.

¡IMPORTANTE! Es necesario copiarse este token, ya que si cierra la página no podrá volver a verlo.

Para clonar repositorios privados o hacer cambios en el repositorio remoto (*push*), puede identificarse de dos maneras:

Token en la URL

Si clone el repositorio utilizando el token, no le pedirá identificarse en ningún momento.

Este método es el más cómodo, pero no es seguro, ya que la PTA se queda guardada en el archivo `.git/config` en la historia de la shell.

La sintaxis del orden es

```
$ git clone https://<token>@github.com/<user>/<repo>.git

# Exemple
$ git clone
https://ghp_3M7ycilbR6fNEGnIsWJyzaoMHVkpXC3jo1ym@github.com/fpmislata-dam1s-ed/git_training.git
```

En el ejemplo utilice la organización fpmislata-dam1s-ed como propietario para realizar el ejemplo. En su caso, debe introducir el voster usuario.

```
jpuigcerver@0A6PC02:~$ git clone https://ghp_3M7ycilbR6fNEGnIsWJyzaoMHVkpXC3jo1ym@github.com/fpmislata-dam1s-ed/git_training.git
Cloning into 'git_training'...
remote: Enumerating objects: 100, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 100 (delta 13), reused 35 (delta 11), pack-reused 62
Receiving objects: 100% (100/100), 1.85 MiB | 3.64 MiB/s, done.
Resolving deltas: 100% (23/23), done.
jpuigcerver@0A6PC02:~$ cd git_training/
jpuigcerver@0A6PC02:~/git_training$ ls
Alice.txt  img  README.md  translations
jpuigcerver@0A6PC02:~/git_training$
```

Token como contraseña

La otra opción es utilizar el PAT como si fuera la contraseña.

```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
> Cloning into `YOUR-REPOSITORY`...
Username: <type your username>
Password: <type your password or personal access token (GitHub)>
```