

## Uso de GIT y GITHUB

**Paso 0:** crear una cuenta en github

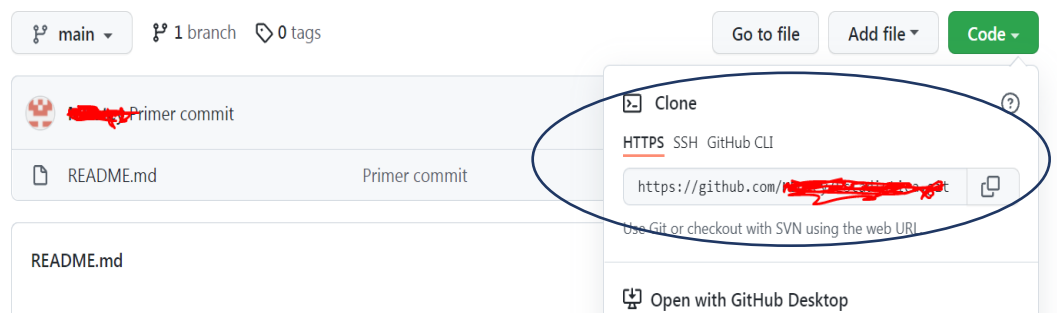
**Paso 1:** instalar git, crear repositorio en github y clonar el repositorio remoto en local

1. Instalar git para los que tengan Windows, los que tengan MAC, abrir consola.
2. Login Github. Crear un nuevo repositorio con el nombre de: "machine\_learning\_supervisado". Este repositorio debe contener README.md en github
3. Abrir *git bash*:
  - Ver en qué carpeta nos encontramos: `pwd`
  - Ir a la ruta donde se desea copiar el repositorio de github. Para moverte entre carpetas por la línea de comandos usa las siguientes sentencias:  
`cd . .` para ir a la carpeta anterior a la que estás  
`cd nombre_carpeta` para ir a la siguiente carpeta contenida en la actual

- Clonar el repositorio remoto de github:

`git clone https://github.com/...`

La ruta https se encuentra situándose en el repositorio creado en github:



A continuación, se puede elegir la opción 1 o opción 2 para configurar git con github.

## Opción 1: subir un cambio al repositorio de github

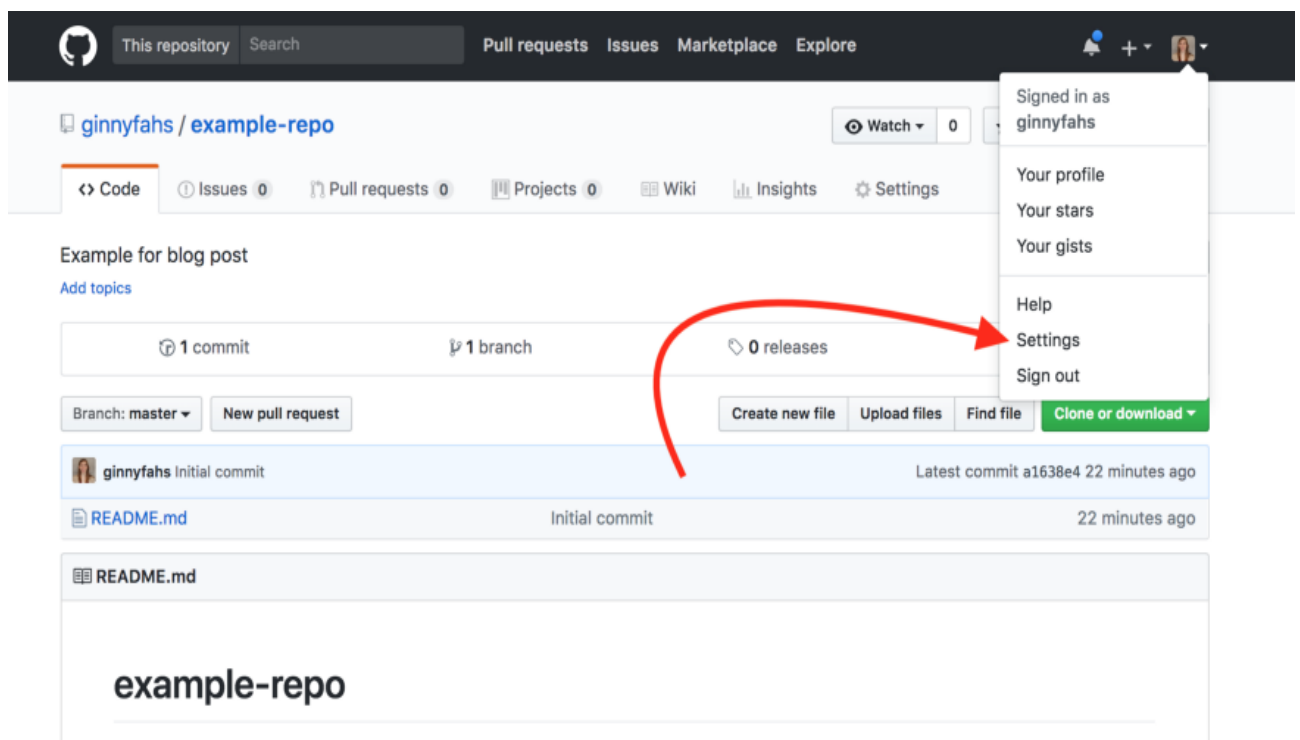
**Paso 1:** configurar el nombre y el correo.

```
git config --global user.email "email"  
git config --global user.name "Nombre"
```

**Paso 2:** configurar la clave de github para introducirla en la consola de git.

Seguir los siguientes pasos para crear una clave:

**Settings => Developer Settings => Personal Access Token => Generate New Token** (Give your password) => **Fillup the form** => click **Generate token** => **Copy the generated**



Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Saved replies

Applications

Developer settings

## Public profile

**Name**

Ginny Fahs

**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.

**URL**

**Company**

You can @mention your company's GitHub organization to link it.

Search GitHub

Pull requests Issues Marketplace Explore

Settings / Developer settings

OAuth Apps

GitHub Apps

Personal access tokens

### No OAuth applications

OAuth applications are used to access the GitHub API. [Read the docs](#) to find out more.

Register a new application

© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

The screenshot shows the GitHub 'Developer settings' page. On the left sidebar, 'Personal access tokens' is selected. The main area is titled 'Personal access tokens' and contains a 'Generate new token' button and a 'Revoke all' button. Below these, a list of generated tokens is shown, each with its permissions and last used date. A red arrow points from the 'Generate new token' button to the first token entry.

Token Name	Permissions	Last used	Actions
admin:org, admin:org_hook, admin:public_key, admin:repo_hook, repo, user	admin:org, admin:org_hook, admin:public_key, admin:repo_hook, repo, user	Last used within the last 2 weeks	Edit Delete
admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, gist, notifications, repo, user	admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, gist, notifications, repo, user	Last used within the last week	Edit Delete
gist, public_repo	gist, public_repo	Never used	Edit Delete
admin:public_key	admin:public_key	Last used within the last 2 months	Edit Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Copiar la clave y guardar.

**Paso 3:** Volver a la consola git bash. Añadir los cambios realizados que queremos subir al repositorio remoto y comitearlos:

```
git add nombre
git commit -m "Introduccion en nombre el cambio X"
git push
```

```
~ :> git push origin my-branch
```

```
Username for 'https://github.com': myusername
```

```
Password for 'https://myusername@github.com': ← Here :)
```

Donde pone Here, se debe poner la contraseña generada en github.

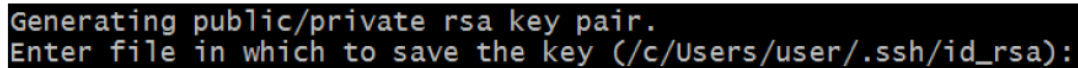
## Opción 2 subir un cambio al repositorio de github

**Paso 1:** generación de claves en git para sincronizar git con github . Abrir la consola de git: *git bash*

1. Generar una nueva clave ssh:

```
ssh-keygen -t rsa -b 4096 -C "email"
```

Cuando se obtenga la siguiente línea:



```
Generating public/private rsa key pair.  
Enter file in which to save the key (/c/Users/user/.ssh/id_rsa):
```

Dar a enter sin introducir nombre de fichero. De esta forma, se guardará el fichero en la ruta por defecto indicada: C:\Users\User\.ssh\id\_rsa

2. Comprobar si se ejecuta el fichero ssh:

```
eval $(ssh-agent -s) (debería dar algo parecido a: Agent PID número)
```

3. Agregar la nueva clave ssh al agente:

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

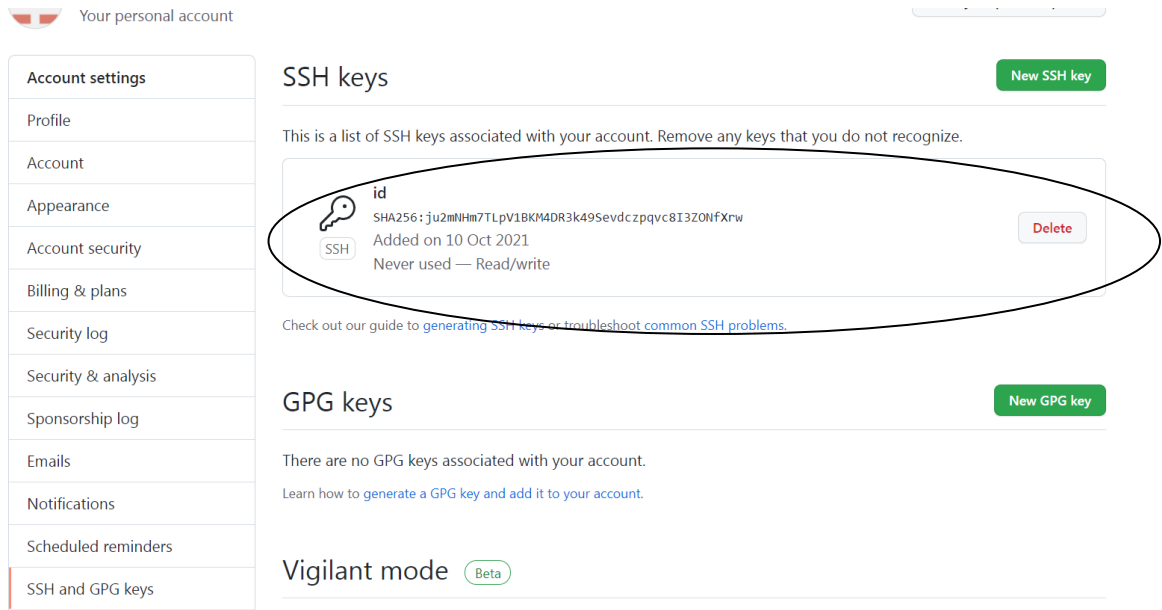
### **Paso 2:**

1. Copiar la clave generada en git a github (en el caso del paso 1 de este Word, la clave se encuentra en el fichero id\_rsa de C:\Users\User\.ssh\id\_rsa.pub)

Cuidado: Importante! coger el fichero .pub y no el que no tiene .pub (la contraseña que se ha de introducir en github es la que empieza por ssh-rsa).

2. Ir a Usuario (arriba a la derecha)→Settings→SSH and GPG keys.

Copiar la clave "ssh-rsa ..." generada en la carpeta del paso 1. Cuidado con no copiar espacios iniciales. Debe ser exactamente igual que la del fichero id\_rsa



The screenshot shows the GitHub 'Your personal account' settings page. On the left is a sidebar menu with options: Account settings, Profile, Account, Appearance, Account security, Billing & plans, Security log, Security & analysis, Sponsorship log, Emails, Notifications, Scheduled reminders, and SSH and GPG keys (which is highlighted). The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below the title is a message: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' A table lists one SSH key with the title 'id', a key ID 'SHA256:ju2mNHm7TLpV18KM4DR3k49Sevdczpqvc8I3Z0NfXrw', the date 'Added on 10 Oct 2021', and the status 'Never used — Read/write'. A 'Delete' button is next to the key. Below the table is a link to a guide: 'Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH problems](#).' Below this is the 'GPG keys' section with a 'New GPG key' button and a message: 'There are no GPG keys associated with your account.' A link says 'Learn how to [generate a GPG key and add it to your account](#).' At the bottom is the 'Vigilant mode' section with a 'Beta' badge.

**Paso 3:** en *git bash*, realizar el primer push.

1. Ir a la carpeta del Proyecto (donde hemos hecho clone en el paso 0)

```
cd ruta/folder/donde/hacer/clone/nombre_repo
git status
git config --global user.email "email"
git config --global user.name "Nombre"
```

2. Añadir los cambios realizados que queremos subir al repositorio remoto y comitearlos:

```
git add nombre
git commit -m "Introduccion en nombre el cambio X"
```

3. Añadir la ruta del repositorio remoto en git:

```
git remote add origin
https://github.com/muy_user_name/my_repo.git
git remote -v
```

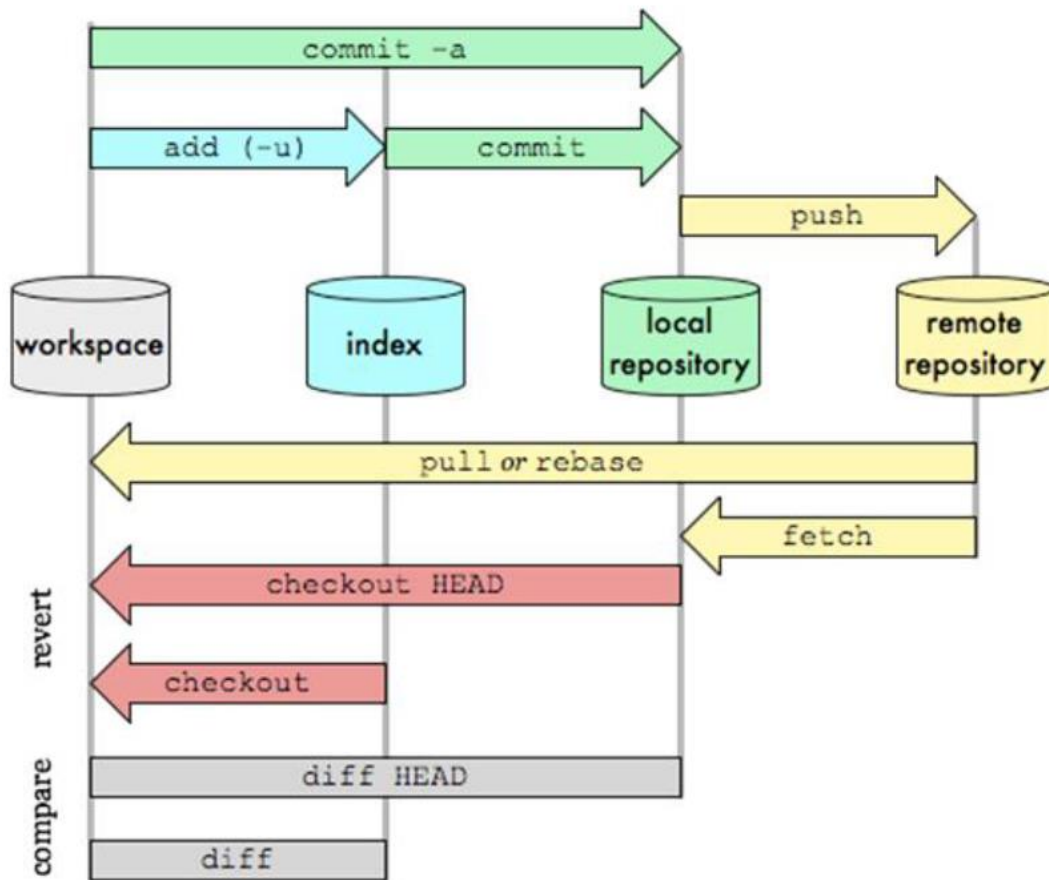
4. Subir al repositorio remote el commit:

```
git push https://github.com/muy\_user\_name/my\_repo.git
```

5. Verificar que en el repositorio de github, se ha subido el cambio que hemos realizado.  
FIN.

## Flujo de git

### Git Data Transport Commands



## Uso de entornos en Python

**Paso 1:** abrir anaconda prompt

**Paso 2:** desde la terminal de anaconda prompt, realizar los siguientes pasos:

- Ver los entornos disponibles actuales

```
conda env list
```

- Instalar la librería que facilita usar kernels en conda dentro del entorno base

```
conda install nb_conda_kernels
```

- Crear un Nuevo entorno asociado a un Proyecto

```
conda create --name "NOMBRE"
```

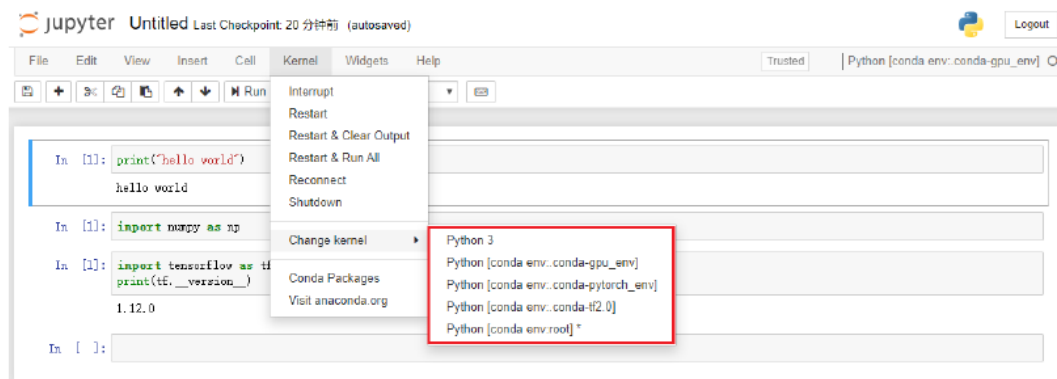
```
conda activate "NOMBRE"
```

```
conda install ipykernel
```

- Crear un kernel de anaconda asociado al entorno

```
python -m ipykernel install --user --name NOMBRE --display-name "NOMBRE"
```

**Paso 3:** abrir un notebook de jupyter y ver que el kernel asociado al entorno está correctamente instalado





## Bibliografía

### Git y Github

<https://platzi.com/tutoriales/1557-git-github/4067-configurar-llaves-ssh-en-git-y-github/>  
<https://gist.github.com/almapase/5ded8e3503f7ef7f01648cfaa7215eb3>  
<https://colaboratorio.net/atareao/developer/2017/git-github-instalar-configurar-crear-primer-repositorio/>  
<https://jdblishak.github.io/2014-09-18-chicago/novice/git/05-sshkeys.html>  
<https://ginnyfahs.medium.com/github-error-authentication-failed-from-command-line-3a545bfd0ca8>

### Entornos y kernels

<https://towardsdatascience.com/get-your-conda-environment-to-show-in-jupyter-notebooks-the-easy-way-17010b76e874>  
[https://ipython.readthedocs.io/en/stable/install/kernel\\_install.html](https://ipython.readthedocs.io/en/stable/install/kernel_install.html)