



**Git** es un lenguaje de programación.

Se usa para trabajar con versiones de ficheros.

Los ficheros son ficheros de texto (no imágenes)

Tenemos un repositorio de trabajo local.

Si arrancamos git en ese repositorio local, Podemos hacer seguimiento de todos los ficheros y que estén dentro del repositorio.

Los ficheros pueden estar dentro de carpetas.

**Github** es una Plataforma commercial (una web).

Existen más webs para trabajar con repositories git (Gitlab, Bitbucket...)

Github es una web donde puedo poner repositories de Código públicos o privados.

Puedo trabajar con Github desde la interfaz gráfica de la web (estando autenticado para mis repositories o sin autenticar con otros públicos no míos). También puedo enlazar mi cuenta de Github con mi portátil. Así puedo trabajar en local con git y puedo subir y bajar los cambios publicados en Github.

Si tienes Windows es recomendable instalar GitBash

# SSH KEYS para enlazar tu terminal con Github

Esto SOLO se hace la primera vez que quieres conectar tu terminal a Github

Abre un terminal bash en PythonAnywhere

Lista directorios ocultos (ls -a)

Ve a .ssh (donde se guardan las claves públicas y privadas)

Si es la primera vez, estará vacío. Si no, estará el fichero con la clave pública y la privada. La privada NO SE ENSEÑA A NADIE. La pública, a quien quieras.

# SSH KEYS para enlazar tu terminal con Github

Ve a tu home (cd ~)

Usa el commando de abajo y acepta varias veces (enter) hasta volver al prompt

Ya tienes generadas una clave pública y una privada

```
ssh-keygen -t rsa
```

Ve al directorio con los ficheros y muestra la pública con el commando cat

# SSH KEYS para enlazar tu terminal con Github

Ve a Github

En tu cuenta – settings – SSH and GPG keys

Añade la clave pública

Ahora cuando quieras subir cambios a tu Github podrás hacerlo desde la terminal

Github es el remoto, en git se llama “origin” y está “arriba”. El local está “abajo”.

# Repo en local

git init → git ya está pendiente de todo

git add mi\_fichero1.txt fichero3.py → git sigue esos ficheros

git commit -m "funciones corregidas" → git hace un commit (foto) del estado del repo.