

# GitHub

- Hecho con

CherryTree 0.35.2, en LMDE 2 Betsy Linux

Instalación desde línea de comandos:

```
apt-get install git
```

Se debe crear una cuenta en github.com usuario y password, y un repositorio

Luego configuramos nuestra cuenta en el sistema local:

```
git config --global user.name "Nombre que quieras mostrar"
```

```
git config --global user.email "correo@electroni.com"
```

Se sustituye --local por --global para que un repositorio sea usado en concreto y no varios o todos con --global

Creamos un directorio en nuestra carpeta principal

```
mkdir github && cd github
```

En nuestro nuevo directorio github clonamos un repositorio o creamos archivos en él:

Para clonar:

```
git clone http:// ruta del repositorio de github
```

Por ejemplo:

```
git clone https://github.com/bakara-e/desaparecidos
```

Si todo está bien, si estamos bien autenticados en el git, pasemos a comprobarlos entrando al directorio raíz de nuestra carpeta en que trabajamos y tecleamos:

```
git config --list
```

Deben estar en la lista desplegada user.name y user.mail

## Luego viene la clave de todo:

Para agregar archivos a nuestra memoria caché, digamos, o virtualmente al repositorio donde lo subiremos.

Es decir, para entrar a la antesala antes de subir necesitamos decirle al sistema lo que subiremos, una especie como de señalización o índice con.

```
git add archivo.xxxx archivo.xxx
```

o bien, todos:

```
git add .
```

Y luego escribimos el comando que enviará estos archivos a la sala de espera del avión, ya para abordar:

(Abre un editor de texto antes -en mi caso: nano- en el cual escribimos algunas observaciones del archivo, para que dé efecto el comando)

```
git commit
```

Para enviar los archivos agregados, primero al índice con addm y luego a la antesala, con commit, al repositorio remoto:

```
git push
```

## Comandos de utilidad:

Agregar un archivo desde línea de comandos:

```
> archivo.txt
```

```
> archivo.php
```

Renombrar fichero:

```
mv fichero.txt nuevofichero.php
```

Borrar:

```
rm fichero.txt
```

Podemos usar nuestro editor de texto para meter información y escribir en ellos:

```
gedit archivo.php
```

## Para obtener una llave de seguridad ssh entre el servidor local y remoto:

```
ls -al ~/.ssh
```

Muestra los archivos .ssh existentes

```
ssh-keygen -t rsa -b 4096 -C "aquipon@tucorreo.com"
```

Crea una nueva llave ssh llamada rsa

En el monitor de nuestra terminal GNU/Linux aparecerá:

```
Enter file in which to save the key (/Users/you/.ssh/id_rsa): [Sólo presiona enter, sin escribir]
```

```
Enter passphrase (empty for no passphrase): [Escribe una contraseña]
```

```
Enter same passphrase again: [Escribe otra vez la contraseña]
```

Luego escribimos:

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

Agregamos:

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

Con nuestro editor de texto, en este caso uso gedit escribimos para abrir el siguiente archivo de clave ssh:

```
gedit ~/.ssh/id_rsa.pub
```

Lo que copiamos de este archivo lo pegamos en nuestro GitHub, primero clic en la imagen de engrane de configuración:



Luego >>

Billing

Payment history

SSH keys

Security

Luego >>

SSH Keys

Add SSH key

There are no SSH keys with access to your account.

Luego pegamos y dejamos en blanco el título >>

## Add an SSH Key

Title

Key

ssh-rsa

AAAAB3NzaC1yc2EAAAABIwAAAQEAx04FILYB7QDkJ5Tlff5lJNPLcYWcVR5xBmXNEVPQ4Wn1A4AjDcfCCqIOb  
Rf2rjxsaHoR1LEV+HUK+HKmNimrcw546T0OtqgnT+sVsQGn+3xqSpjtla+2xbZiy4eWrbNzarcQfcysMStVRBKBryDj  
rGu3qR96TUxGewYaa7Yb4qHHHSLYMTI6ac7Qk7WMvcbB2CMYNWnlbW6mEvO+FzizshnBku5ii4mE+NDSSev  
mytzgZRQQg9DOVrt7BOKPpOJ5J3Ej19rFzCiTMCdBWHxzH3s8+01Aqell++KaUuqHGycgtg2uYICakWLvdCreW  
Yd3iqPV09uYq/HarH4yuOBHaw== tekkub@gmail.com

Agregamos la llave a nuestro servidor de GitHub >>

Add key

Listo, nuestros servidores local y remoto utilizarán esta llave ssh para comunicarse. Es decir, nuestro enlace de seguridad es excelente.