

# Prueba práctica de Git y GitHub de María del Rocío Romo García:

Se pide documentar **en un PDF** la instalación, configuración y puesta en marcha de:

- Git NUEVO en la carpeta raíz de nuestro proyecto futuro (odoo17). Siguiendo la misma estructura de carpetas que usamos para el proyecto odoo16 (ver nota)

```
manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17
$ git init
Initialized empty Git repository in C:/Users/manana/Desktop/Rocio Romo/Odoo17/.git/

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ git config --local user.name "Romo Estudio"

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ git config --local user.email "romocedeco@gmail.com"

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Rocio Romo
user.email=bloomblaze7896@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
user.name=Romo Estudio
user.email=romocedeco@gmail.com
```

Voy explicando lo que está ocurriendo por partes:

1. Git init: Se ha inicializado un nuevo repositorio en una carpeta llamada "Odoo17".
2. git config --local user.name "Romo Estudio": Se ha configurado el nombre de usuario para poder identificar los cambios que haga este usuario en concreto.
3. git config --local user.email "Romo Estudio": Se ha configurado el email del usuario para poder vincularlo con el GitHub del usuario.
4. git config --list: Lanzó este comando para poder identificar los cambios que se han creado. Aparece el usuario (nombre y correo) al final de la captura del video.

```
manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ git config user.name
Romo Estudio
```

He querido verificar el nombre de usuario que se está usando en la carpeta Odoo17.

- NOTA sobre la estructura de carpetas: Solo las carpetas + solo el archivo README.md, que deberá ser explicativo para este proyecto. Así habrá un archivo que poder controlar en el repo y sincronizar con github.

Comandos usados:

- mkdir: Crea carpetas vacías
- echo "algo" > documento: Crea el documento que le digamos con el formato que necesitemos y escribe dentro lo que queramos.

```
manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ mkdir config_odoo

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ mkdir custom-addons

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ mkdir logs
```

Se ha creado la estructura de carpetas.

```
manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ echo "# Práctica evaluable de Odoo17" > README.md

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ echo "; Aquí irán los archivos ocultos" > .gitignore

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (master)
$ ls -a
./ ../ .git/ .gitignore README.md config_odoo/ custom-addons/ logs/
```

Se han creado los archivos ".gitignore" y el "README.md" del proyecto. Por último se ha lanzado el comando ls -a para ver la estructura.

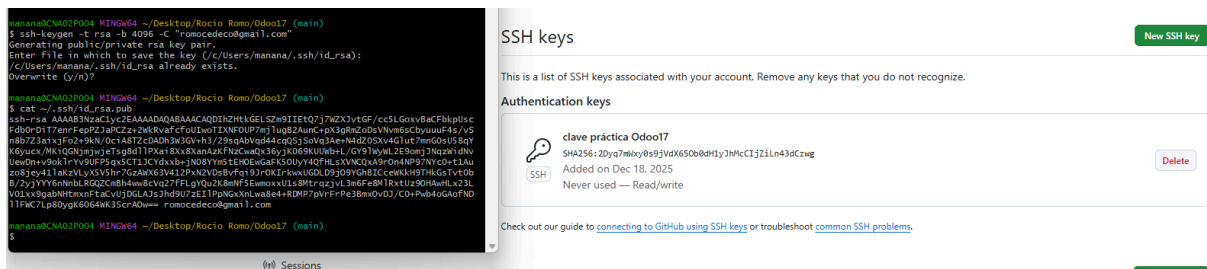
- NOTA sobre las variables de entorno de GIT: no deben ser "--global" deberán ser "--local", para no pisarse con las variables globales que ya tenemos definidas en nuestro Git anterior.
- Hacer lo propio en remoto con Github. Cuenta NUEVA de github. Para lo que hará falta **OTRO** correo y **OTRO** nombre de usuario !!!! QUE SERÁN LOS QUE HAYAMOS CONFIGURADO CON "**git config --local user.name y user.email**" en la configuración del punto anterior de Git.

Previamente se ha creado el usuario, adjunto el enlace a la cuenta (<https://github.com/StudioRomo>) y al repositorio de esta práctica (<https://github.com/StudioRomo/Odoo17>)

- Vincular los dos repos para poder trabajar sincronizados. El local y el remoto

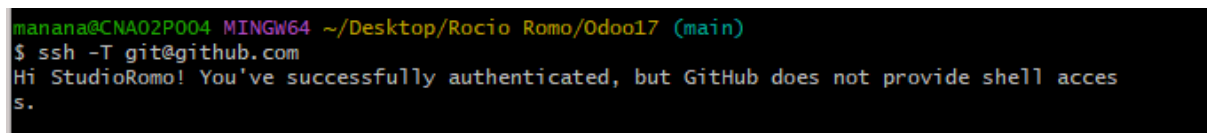
Para ello voy a crear una clave SSH, pero voy a usar RSA ya que de la forma vista en clase me la sobreescribe.

1. Se crea la clave con: `ssh-keygen -t rsa -b 4096 -C "romocedeco@gmail.com"` (Hay que darle a enter 3 veces)
2. Lanzamos el comando: `cat ~/.ssh/id_rsa.pub` para obtener la clave y la pegamos en los ajustes de GitHub cuando creamos la clave SSH.



Sé que ha habido compañeros que han modificado la ruta dónde se ha almacenado la clave pero he preferido hacerlo así.

Voy a probar la conexión con el GitHub usando `ssh -T git@github.com` y vemos que efectivamente funciona por que nos saluda.



Tras todo esto procedo a subir los archivos.

```
manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (main)
$ git remote remove origin

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (main)
$ git remote add origin git@github.com:StudioRomo/Odoo17.git

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (main)
$ git branch -M main

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 336 bytes | 336.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:StudioRomo/Odoo17.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

manana@CNA02P004 MINGW64 ~/Desktop/Rocio Romo/Odoo17 (main)
$
```

He tenido que poner el comando "git remote remove origin" por que al principio me estaba dando problemas con la subida, tras quitarlo he podido vincular el repositorio ( git remote add origin git@github.com:StudioRomo/Odoo17.git), he cambiado el nombre de la rama (git branch -M main) y por último subido los cambios ( git push -u origin main )

- La actividad consiste en hacer todo lo que ya hemos hecho antes para el Git de la carpeta odoo16.