

Hoja de Trabajo No 1

Juan Carlos Romero

Carnet: 21006035

Definición Git es un sistema de control de versiones para control edición y seguimiento de cambios en códigos que facilita la colaboración entre un equipo de trabajo.

Metadatos Git mantiene en archivos llamados metadatos la información necesarias para identificar la versión, los mensajes del cambio realizado y quien realizó los cambios etc.

Metadatos

1. Ir a <https://git-scm.com/>
2. Buscar la sección descargas
3. Seleccionar la versión compatible con nuestro sistema operativo
4. Después de instalar verificar la versión

Comandos

Version de Git

```
In [42]: %system git version
```

```
Out[42]: ['git version 2.21.0.windows.1']
```

Comando de ayuda

```
In [43]: %system git help --version
```

```
Out[43]: ["error: unknown option `version'",
'usage: git help [--all] [--guides] [--man | --web | --info] [<command>]',
'',
',
' -a, --all          print all available commands',
' -g, --guides       print list of useful guides',
' -c, --config       print all configuration variable names',
' -m, --man          show man page',
' -w, --web          show manual in web browser',
' -i, --info         show info page',
' -v, --verbose      print command description',
'']
```

Iniciando con GIT

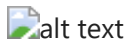
Create directorio Creamos el directorio donde almacenaremos nuestro proyecto GIT

1. Abrir cmd
2. Crear directorio de trabajo 2.1. *MD Tarea1_Jupyter_Notebooks_Markdown_Git*
3. Ir al directorio de trabajo 2.2. *CD Tarea1_Jupyter_Notebooks_Markdown_Git*

Iniciando el repositorio

```
In [1]: %system git init
```

```
Out[1]: ['Initialized empty Git repository in C:/Users/CES-UVG/Documents/NewroX/Maestria DataScience/Ciencia de Datos en Python/Tarea1_Jupyter_Notebooks_Markdown_Git/.git/']
```



Verificando el status de nuestro repositorio

Por ahora no hemos agregado ningun archivo para seguimiento, podemos verificar el estado de los archivo ejecutando el comando status -s

```
In [3]: %system git status -s
```

```
Out[3]: ['?? .ipynb_checkpoints/',
        '?? Carpeta_init_git.JPG',
        '?? Practicando Markdown y Git.ipynb']
```

Agregando nuestros archivos al area local de nuestro repositorio

El simbolo ?? nos indica que los archivos no han sido agregados al area temporal. Ahora procederemos a agregarlos para seguiminto con el comando "add"

```
In [6]: %system git add Carpeta_init_git.JPG
```

```
Out[6]: []
```

```
In [7]: %system git add "Practicando Markdown y Git.ipynb"
```

```
Out[7]: []
```

```
In [8]: %system git status -s
```

```
Out[8]: ['A ".ipynb_checkpoints/Practicando Markdown y Git JC-checkpoint.ipynb"',
        'A ".ipynb_checkpoints/Practicando Markdown y Git-checkpoint.ipynb"',
        'A Carpeta_init_git.JPG',
        'A "Practicando Markdown y Git.ipynb"']
```

Al verificar el estado de los archivos nos aparece como **"A"** que significa que estan agregados. Los Archivos se han agregado al area local/temporal de nuestro repositorio.

Tomando una instantanea de nuestros archivos

Para hacer la primera copia (version) utilizamos el comando **Commit** seguido de la copia -m donde indicamos un mensaje para identificar el porque estamos creando esta version.

```
In [9]: %system git commit -m "Aguemos por primera vez los archivo de nuestro proyecto"
```

```
Out[9]: ['[master (root-commit) 2f81e6d] Aguemos por primera vez los archivo de nuestro project
o',
        ' 4 files changed, 492 insertions(+)',
        ' create mode 100644 .ipynb_checkpoints/Practicando Markdown y Git JC-checkpoint.ipyn
b',
        ' create mode 100644 .ipynb_checkpoints/Practicando Markdown y Git-checkpoint.ipynb',
        ' create mode 100644 Carpeta_init_git.JPG',
        ' create mode 100644 Practicando Markdown y Git.ipynb']
```

Ahora verificaremos el nuevo estado de nuestros archivos

```
In [11]: %system git status -s
```

```
Out[11]: [' M "Practicando Markdown y Git.ipynb"']
```

Agregando otro archivo

1. Agregaremos un nuevo archivo a nuestro folder
2. Agregaremos el archivo a git al area temporal
3. Tomaremos una instantanea con el nuevo archivo

```
In [47]: %system echo About The Project Este es una practica para usar Git AUTOR Juancho Romero
```

```
Out[47]: []
```

```
In [48]: %system git status -s
```

```
Out[48]: [' M "Practicando Markdown y Git.ipynb"', '?? README.txt']
```

- Con el commando *add* agregamos todos los archivos para seguimiento*

```
In [49]: %system git add .
```

```
Out[49]: ['warning: LF will be replaced by CRLF in Practicando Markdown y Git.ipynb.',  
          'The file will have its original line endings in your working directory']
```

```
In [50]: %system git status -s
```

```
Out[50]: ['M "Practicando Markdown y Git.ipynb"', 'A README.txt']
```

```
In [51]: %system git commit -m "Agregamos el archivo nuevo Readme.txt "
```

```
Out[51]: ['[main f4ef246] Agregamos el archivo nuevo Readme.txt',  
          ' 2 files changed, 51 insertions(+), 12 deletions(-)',  
          ' create mode 100644 README.txt']
```

Modificando un archivo

1. Modificamos el nuevo archivo a nuestro folder
2. Agregaremos el archivo a git al area temporal
3. Tomaremos una instantanea con la modificacion de archivo

```
In [29]: %system echo About The Project Este es una practica para usar Git AUTOR Juancho Romero
```

```
Out[29]: []
```

```
In [52]: %system git add .
```

```
Out[52]: []
```

```
In [31]: %system git status -s
```

```
Out[31]: ['M "Practicando Markdown y Git.ipynb"', 'M README.txt']
```

```
In [32]: %system git commit -m "Agregamos la fecha en el archivo Readme"
```

```
Out[32]: ['[master 87213ee] Agregamos la fecha en el archivo Readme',  
          ' 2 files changed, 42 insertions(+), 12 deletions(-)']
```

Conociendo las instantaneas que hemos realizado

Para conocer el listado de commit realizados utilizamos el comando `log --oneline`

```
In [33]: %system git log --oneline
```

```
Out[33]: ['87213ee Agregamos la fecha en el archivo Readme',  
          '3443619 Agregamos el archivo Readme.txt',  
          'f1bdc79 Agregamos el archivo Readme.txt',  
          '2f81e6d Agregamos por primera vez los archivo de nuestro proyecto']
```

Regresando a una copia anterior

Para ir a una copia anterior debemos utiliza el comando **git reset --hard identificador** * El identificador es el que nos aparece en el listado del log

```
In [34]: %system git reset --hard f1bdc79
```

```
Out[34]: ['HEAD is now at f1bdc79 Agregamos el archivo Readme.txt']
```

Hemos retrocedido en el tiempo!!!!


Ahora queremos regresar a la ultima version, para ello ejecutamos el mismo comando con la ultima version del commit

```
In [35]: %system git reset --hard 2f81e6d
```

```
Out[35]: ['HEAD is now at 2f81e6d Agregamos por primera vez los archivo de nuestro proyecto']
```

Estamos en el presente!!!!

Subiendo nuestro repositorio a la nube

1. Para este paso crearemos una cuenta en GitHub [Link](#)
2. Seleccionamos la opcion crear nuevo repositorio 
3. Asignamos un nombre en este caso yo le asigne: "HolaMundo_Jupyter_Markdown_Git_JC"
4. buscamos el nombre del repositorio
5. Ejecutamos el comando **git remote add origin**, Para indicarle a Git que va a trabajar de forma remota. (Esto solo se hace una vez)
6. Para sincronizar los commit locales en la nube ejecutamos el comando **git push -u origin master** *Nota es posible que nos pida el usuario y contraseña de nuestro GitHub*

```
In [56]: %system git remote add origin https://github.com/JuanNewroX/Actividad_Jupyter_Markdown_
```

```
Out[56]: ['fatal: remote origin already exists.']
```

```
In [57]: %system git add .
```

```
Out[57]: ['warning: LF will be replaced by CRLF in Practicando Markdown y Git.ipynb.',  
          'The file will have its original line endings in your working directory']
```

```
In [58]: %system git commit -m "Commit para Github Para entregar"
```

```
Out[58]: ['[main 4562f0c] Commit para Github Para entregar',  
          ' 1 file changed, 29 insertions(+), 28 deletions(-)']
```

In [59]: `%system git push -u origin main`

Out[59]: ["Branch 'main' set up to track remote branch 'main' from 'origin'.",
'To https://github.com/JuanNewroX/Actividad_Jupyter_Markdown_Git_JC.git',
' f4ef246..4562f0c main -> main']

In []: