



# CTE-334. Desarrollo de Aplicaciones de Sistemas de Información Geográfica

**José D. Cáceres**

Licenciatura en Ciencia y Tecnologías de la Información Geográfica

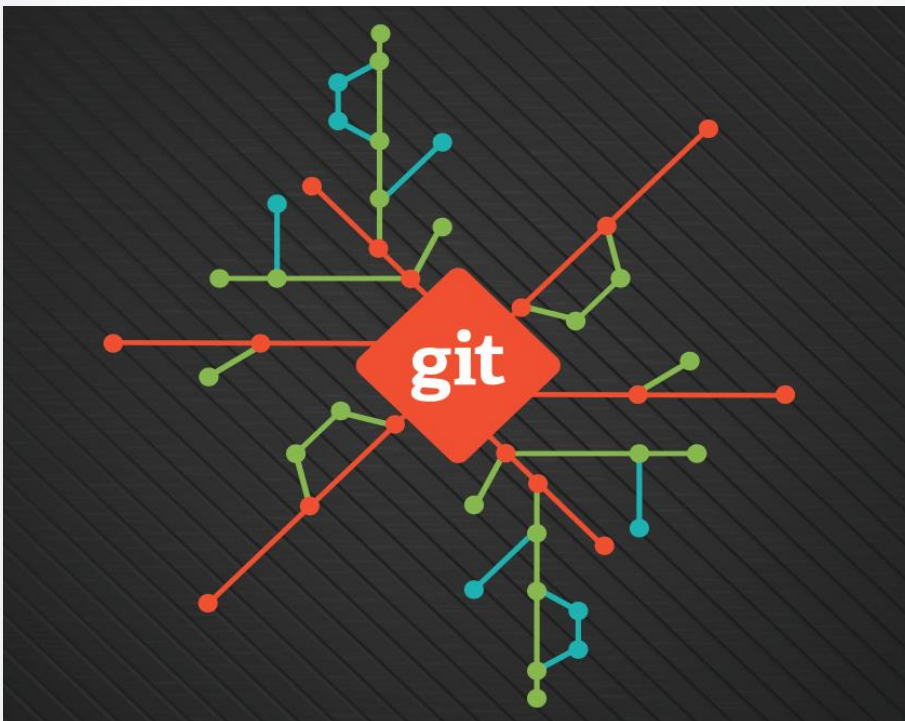
Facultad de Ciencias Espaciales



**UNAH**

# UNIDAD I

## 1. INTRODUCCIÓN A GIT



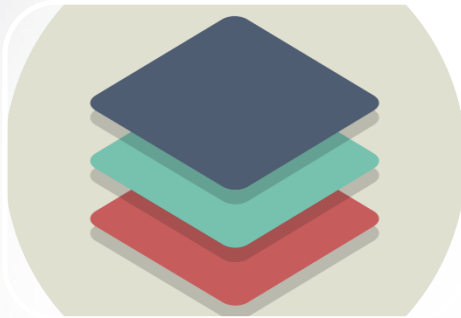
**UNAH**

# Git

---

- Es un sistema de control de versiones distribuido para rastrear cambios en cualquier conjunto de archivos, originalmente diseñado para coordinar el trabajo entre programadores que cooperan en el código fuente durante el desarrollo de software.
- Creado por Linus Torvald.
- Git es un software gratuito y de código abierto

# Git



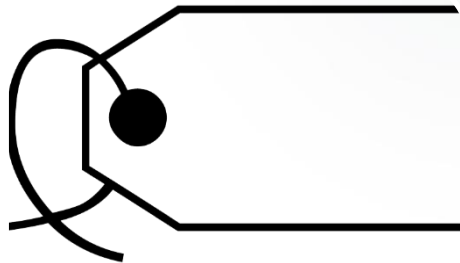
Control de  
Versiones



Auditoría de código



Control de cambios  
en el tiempo



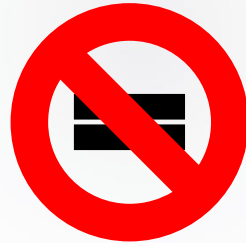
Etiquetas de  
versiones



Trabajo  
colaborativo

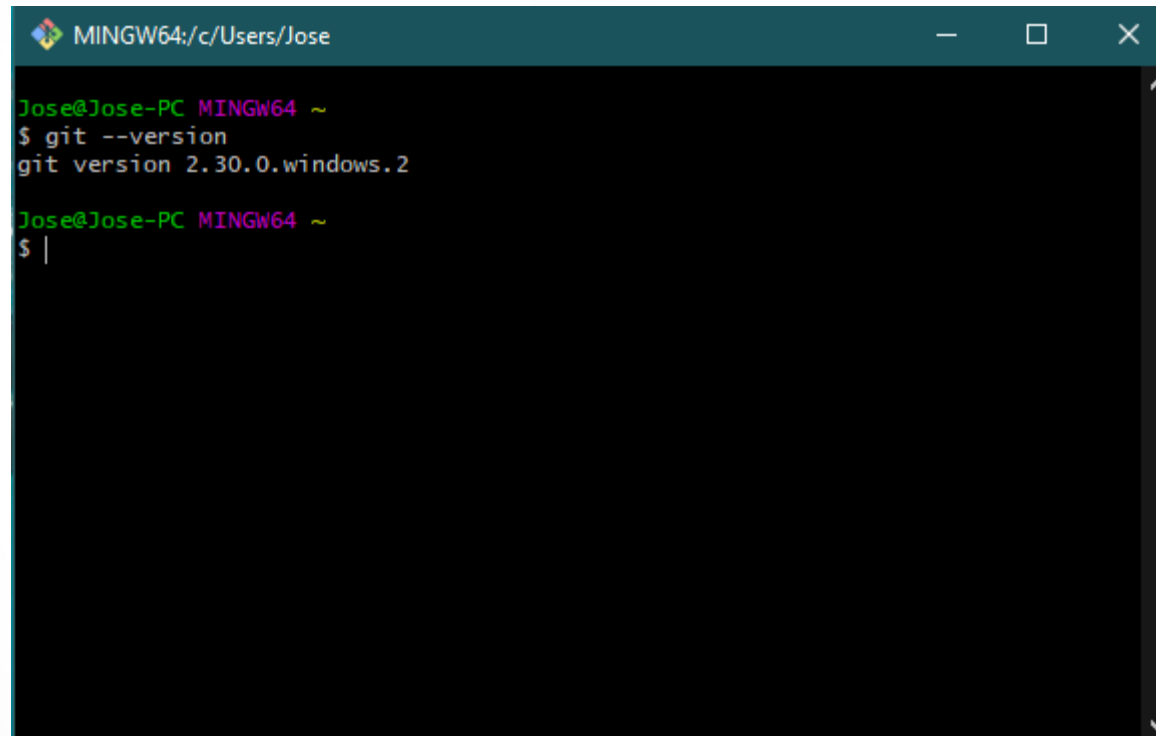
# Git

---



# Comenzando con Git

*\$ git --version*



```
MINGW64:/c/Users/Jose
Jose@Jose-PC MINGW64 ~
$ git --version
git version 2.30.0.windows.2
Jose@Jose-PC MINGW64 ~
$ |
```

# Comenzando con Git

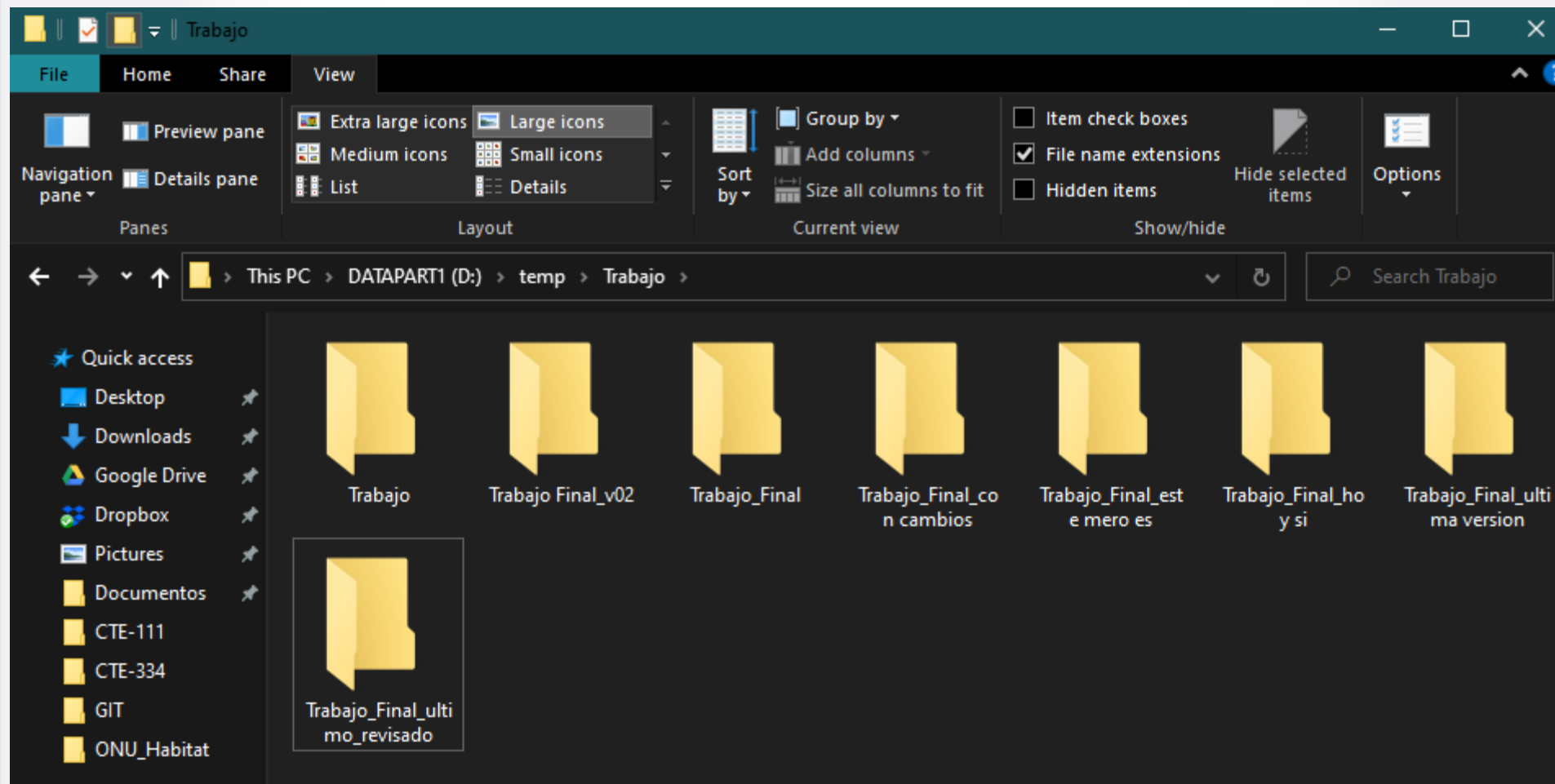
---

- Definir valores de la persona que está haciendo cambios:

*\$ git config --global user.name "SuNombre"*

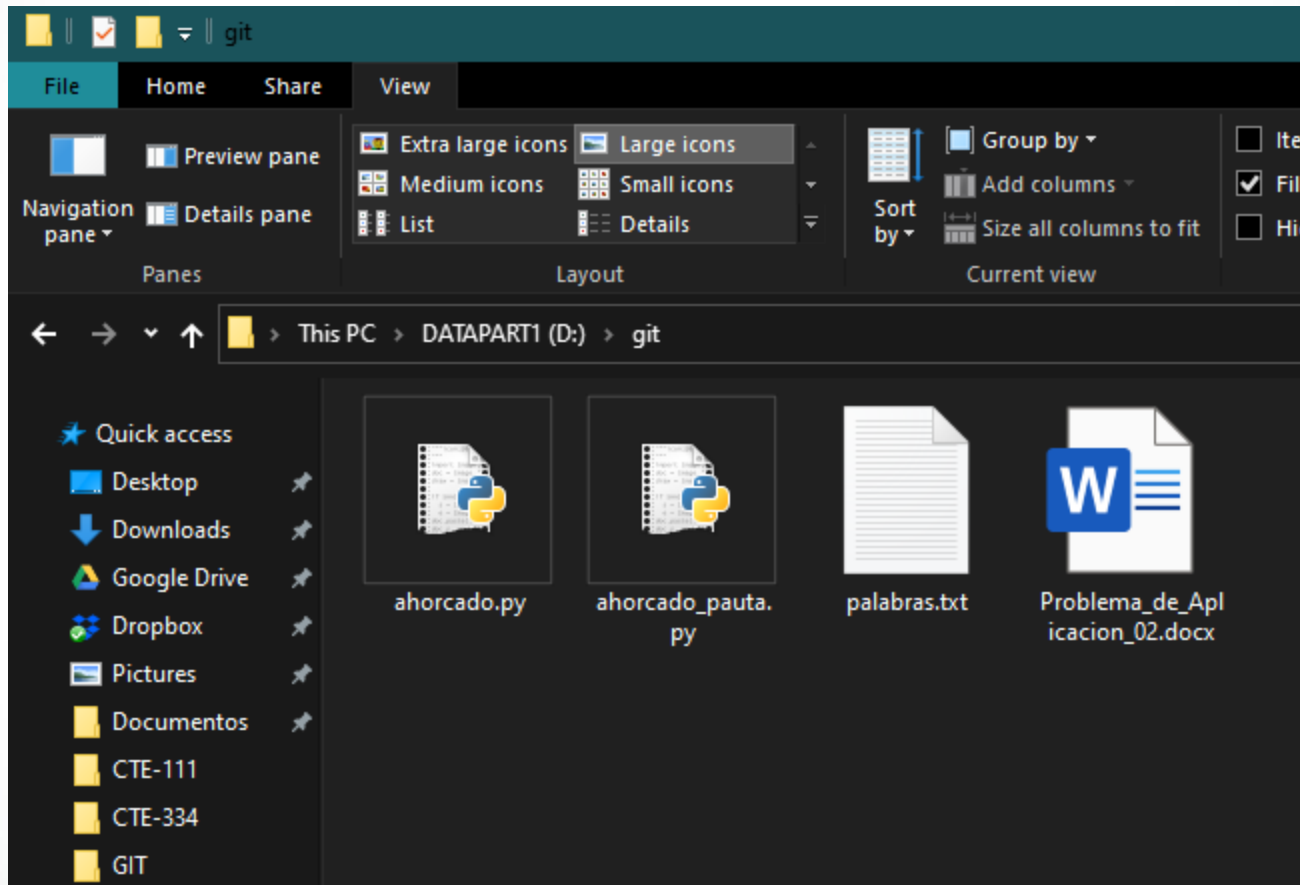
*\$ git config --global user.email "SuEmail@dominio"*

# Utilidad de Versionamiento



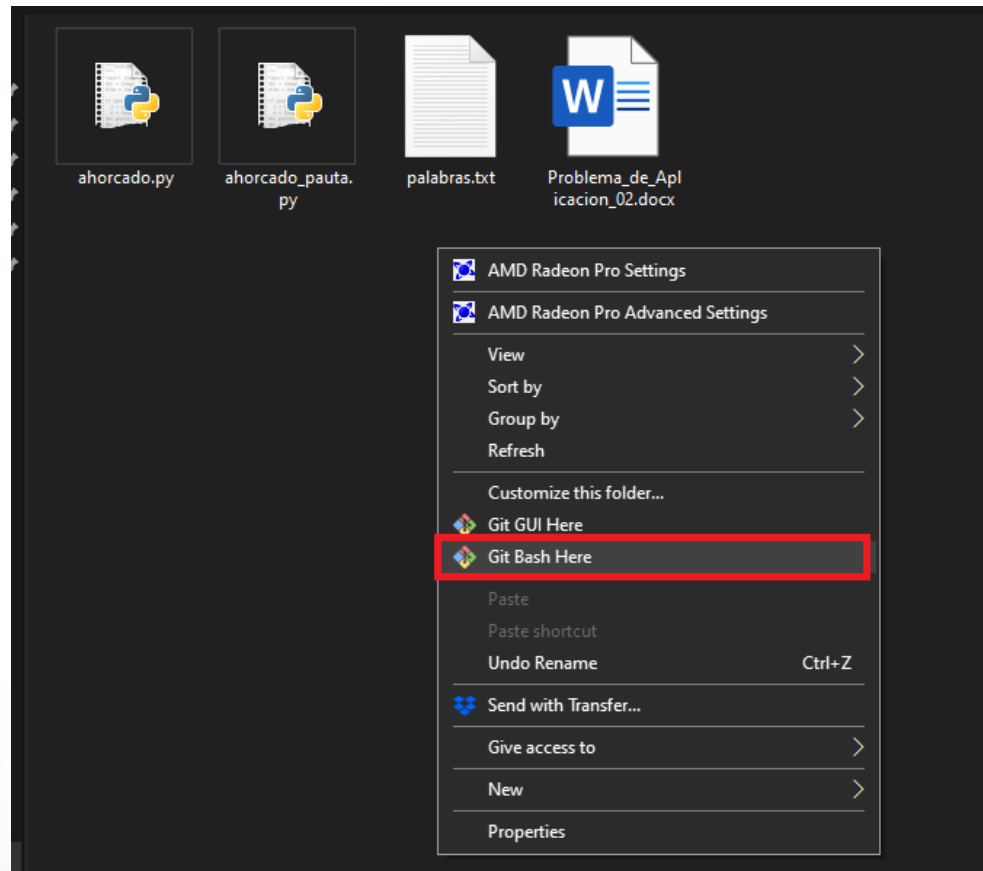


# Mi primer repositorio



# Mi primer repositorio

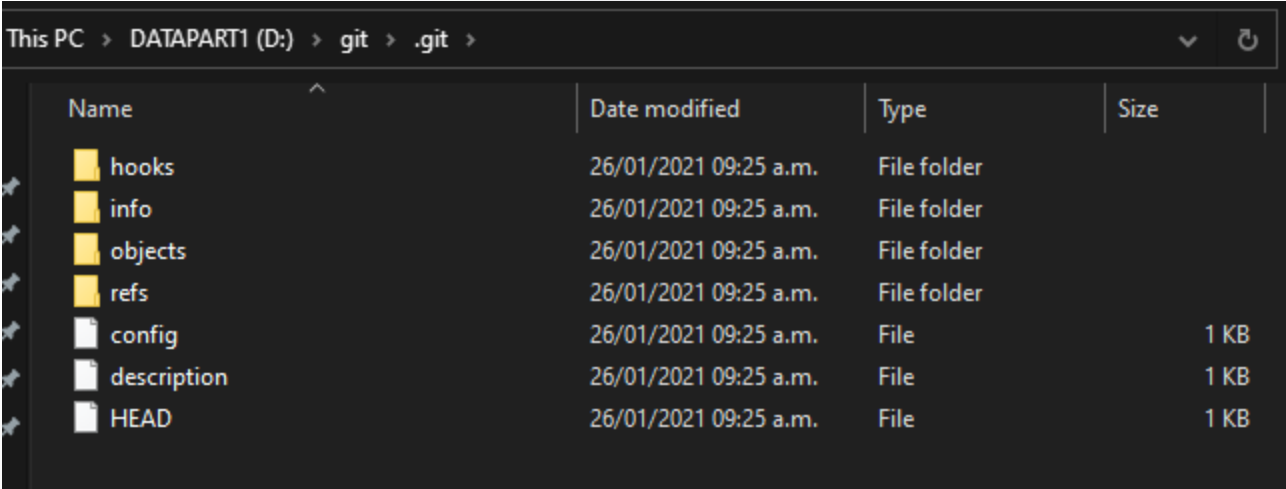
- *Lanzar Git Bash desde la carpeta de trabajo*



# Mi primer repositorio

- *Convertir el espacio de trabajo a un repositorio git*

*\$ git init*

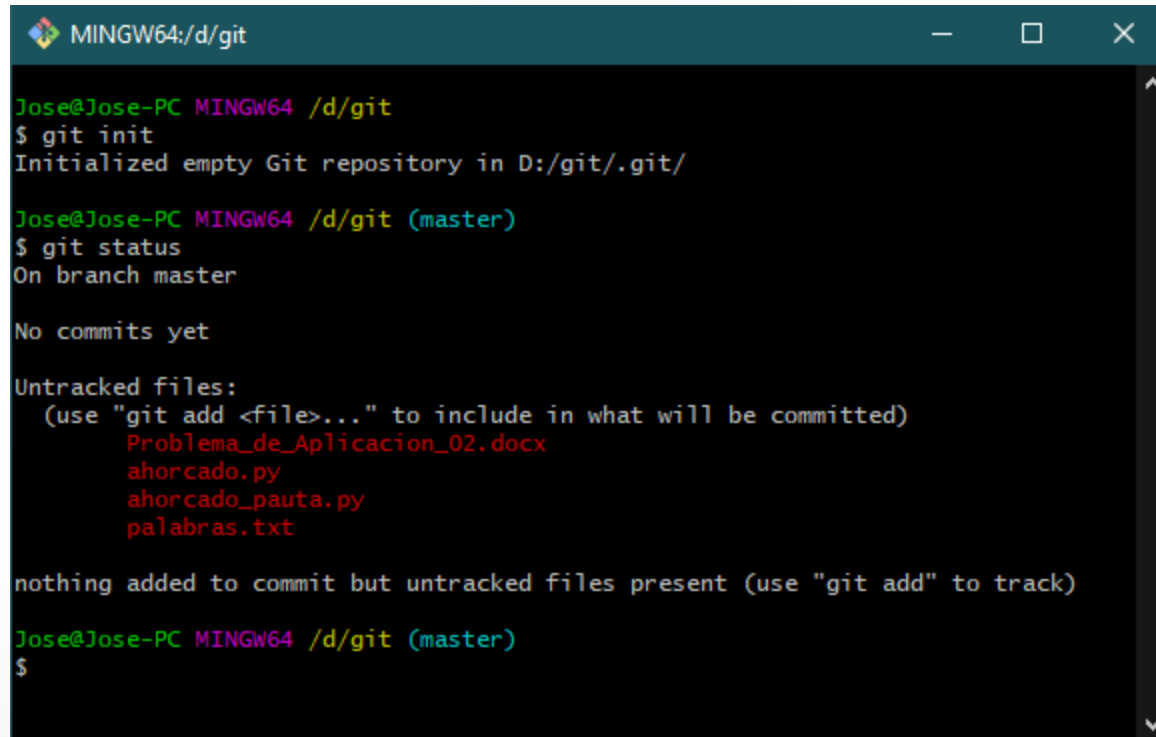


Name	Date modified	Type	Size
hooks	26/01/2021 09:25 a.m.	File folder	
info	26/01/2021 09:25 a.m.	File folder	
objects	26/01/2021 09:25 a.m.	File folder	
refs	26/01/2021 09:25 a.m.	File folder	
config	26/01/2021 09:25 a.m.	File	1 KB
description	26/01/2021 09:25 a.m.	File	1 KB
HEAD	26/01/2021 09:25 a.m.	File	1 KB

# Mi primer repositorio

- *Consultar el estado de los archivos*

*\$ git status*

A screenshot of a MINGW64 terminal window with a dark background. The window title is 'MINGW64:/d/git'. The terminal shows the following commands and output:

```
Jose@Jose-PC MINGW64 /d/git
$ git init
Initialized empty Git repository in D:/git/.git/

Jose@Jose-PC MINGW64 /d/git (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Problema_de_Aplicacion_02.docx
        ahorcado.py
        ahorcado_pauta.py
        palabras.txt

nothing added to commit but untracked files present (use "git add" to track)

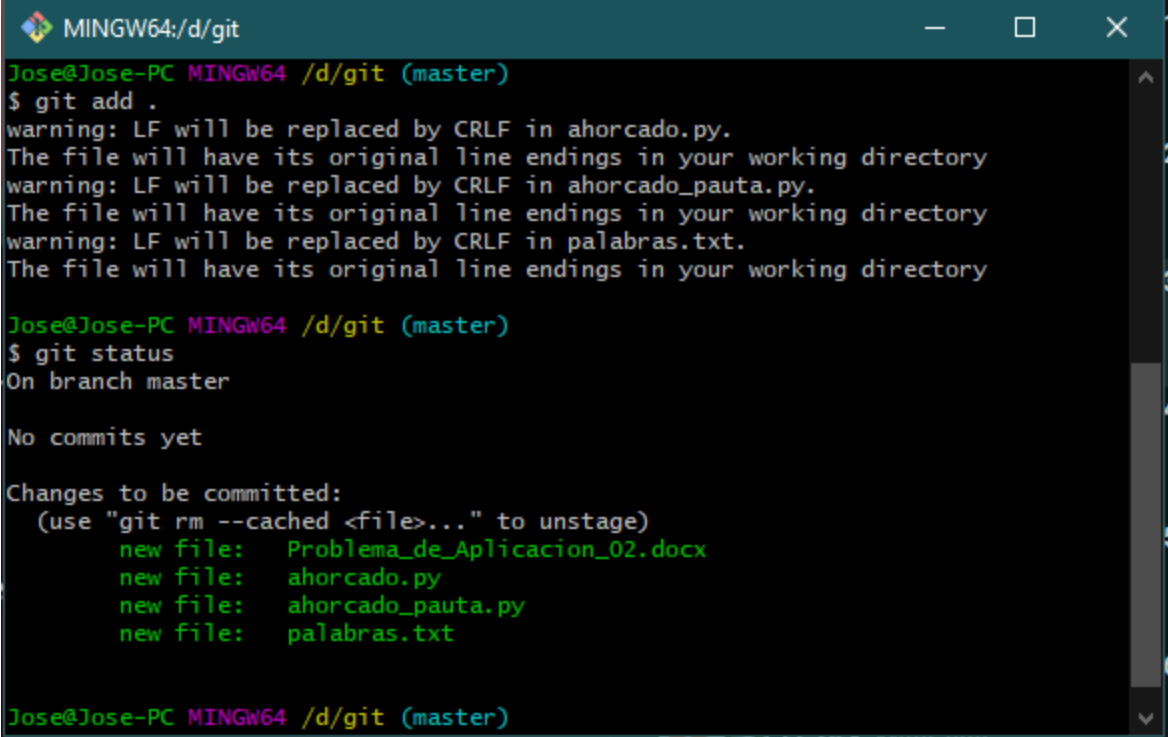
Jose@Jose-PC MINGW64 /d/git (master)
$
```

# Mi primer repositorio

- *Permitir a git que haga el seguimiento de cambios*

*\$ git add .*

*\$git status*

A screenshot of a terminal window titled 'MINGW64:/d/git'. The prompt is 'Jose@Jose-PC MINGW64 /d/git (master)'. The user enters '\$ git add .' and the output shows three warnings about CRLF line endings being replaced by LF in 'ahorcado.py', 'ahorcado\_pauta.py', and 'palabras.txt'. The user then enters '\$ git status' and the output shows 'On branch master', 'No commits yet', and a list of 'Changes to be committed' including 'Problema\_de\_Aplicacion\_02.docx', 'ahorcado.py', 'ahorcado\_pauta.py', and 'palabras.txt'.

```
MINGW64:/d/git
Jose@Jose-PC MINGW64 /d/git (master)
$ git add .
warning: LF will be replaced by CRLF in ahorcado.py.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in ahorcado_pauta.py.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in palabras.txt.
The file will have its original line endings in your working directory

Jose@Jose-PC MINGW64 /d/git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Problema_de_Aplicacion_02.docx
    new file:   ahorcado.py
    new file:   ahorcado_pauta.py
    new file:   palabras.txt

Jose@Jose-PC MINGW64 /d/git (master)
```

# Mi primer repositorio

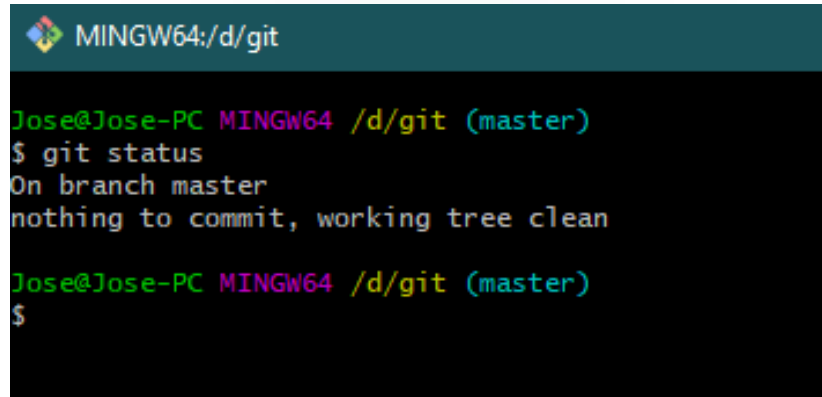
---

- *Crear un mensaje de los cambios realizados*  
*\$ git commit -m "Mensaje"*

```
Jose@Jose-PC MINGW64 /d/git (master)
$ git commit -m "First commit"
[master (root-commit) 544e699] First commit
4 files changed, 79389 insertions(+)
create mode 100644 Problema_de_Aplicacion_02.docx
create mode 100644 ahorcado.py
create mode 100644 ahorcado_pauta.py
create mode 100644 palabras.txt
```

# Mi primer repositorio

- *Ya git tiene control de nuestros archivos*  
*\$ git status*

A screenshot of a Windows command prompt window titled 'MINGW64:/d/git'. The prompt shows a user named 'Jose' at a machine named 'Jose-PC' in the 'MINGW64' environment, currently on the 'master' branch. The user has entered the command '\$ git status'. The output of the command is displayed on the next two lines: 'On branch master' and 'nothing to commit, working tree clean'. The prompt then shows the user entering another '\$' character, indicating the command has finished and the prompt is ready for the next input.

```
MINGW64:/d/git

Jose@Jose-PC MINGW64 /d/git (master)
$ git status
On branch master
nothing to commit, working tree clean

Jose@Jose-PC MINGW64 /d/git (master)
$
```

# Mi primer repositorio

---

- Ver los cambios realizados a los archivos

*\$ git diff*

- Revertir cambios

*\$ git checkout .*



# Mi primer repositorio

---

- Añadir archivos al stage:

```
$ git add <filename>
```

```
$ git add <Dir>/
```

```
$ git add *.fileType
```

- Remover archivos del stage

```
$ git reset <filename>
```

# Mi primer repositorio

---

- Rectificar un mensaje:

```
$ git commit --amend -m "Mensaje"
```

- Renombrar un archivo:

```
$git mv <old_name> <new_name>
```

- Eliminar archivos:

```
$ git rm <filename>
```

*Recuperar un archivo eliminado por git:*

```
$git reset --soft <commit id>
```

```
$git reset --hard <commit id>
```

```
$git reset --mixed <commit id>
```

# Mi primer repositorio

---

- Archivos que no quiero hacer seguimiento deben añadirse a `.gitignore`

# Mi primer repositorio

---

- Crear una nueva rama:

*\$ git branch <name>*

- Cambiar a nueva rama

*\$ git checkout <branch\_name>*

- Unir ramas

*\$ git merge <branch\_name>*

- Eliminar una rama

*\$ git branch -d <branch\_name>*

- Listar todas las ramas

*\$ git branch -a*

# Mi primer repositorio

---

- Subir los archivos a Github
  - Crear un repositorio nuevo en Github  
*\$ git remote add origin <repository\_URL>*
  - Listar repositorios remotos  
*\$ git remote -v*
  - Subir los cambios a Github  
*\$ git push origin <branch\_name>*
  - Eliminar rama remota  
*\$ git push origin --delete <branch\_name>*
  - Eliminar repositorio remoto  
*\$ git remote rm <name>*

# Mi primer repositorio

---

- Descargar los archivos de Github a un repositorio local
  - *Descargar la rama principal*  
*\$ git pull <repository\_URL>*
  - *Descargar una rama específica*  
*\$ git pull <repository\_URL>*  
*<remote\_brach>:<local\_branch>*
  - *Descargar todas las versions*  
*\$ git clone <repository\_URL>*

# COMANDOS COMUNES

- \$ git init
- \$ git config
- \$ git add
- \$ git commit
- \$ git status
- \$ git diff
- \$ git reset
- \$ git checkout
- \$ git branch
- \$ git merge
- \$ git remote
- \$ git push
- \$ git pull
- \$ git log

