



Breve introducción a Git y GitHub





Git

Git

“Git es un *software* de **control de versiones** diseñado por Linus Torvalds, pensando en la **eficiencia**, la **confiabilidad** y **compatibilidad** del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.”

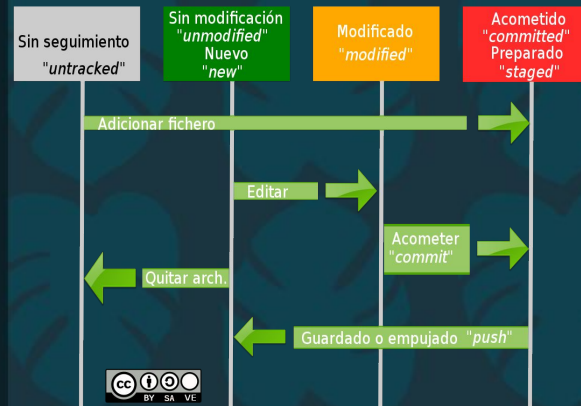
Fuente.

Git. (19 de jun de 2022). En *Wikipedia*.

<https://es.wikipedia.org/w/index.php?title=Git&oldid=144294935>



Ciclo de vida de los archivos mediante Git



Instalación local

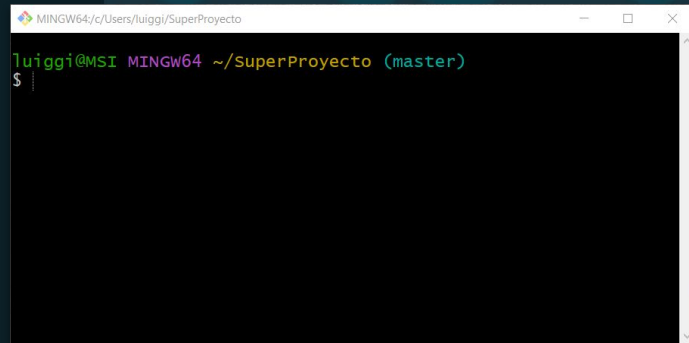
Aunque todo lo haremos en la plataforma Macti, es importante que se familiarice con el uso básico de Git. Para ello haga lo siguiente:

- El software para instalar Git en una computadora local se puede descargar del siguiente sitio <https://git-scm.com/downloads>.
- Existen versiones para cada sistema operativo (SO), elija la que más le convenga.
- Por simplicidad, en MacOS se recomienda usar el *Binary installer* y para Windows utilice una de las siguientes *32-bit Git for Windows Setup* o *64-bit Git for Windows Setup* de acuerdo con su SO.
- Para **Linux** elija de acuerdo al tipo de distribución que tenga (por ejemplo para Ubuntu utilice: `apt-get install git`).

Terminal de comandos

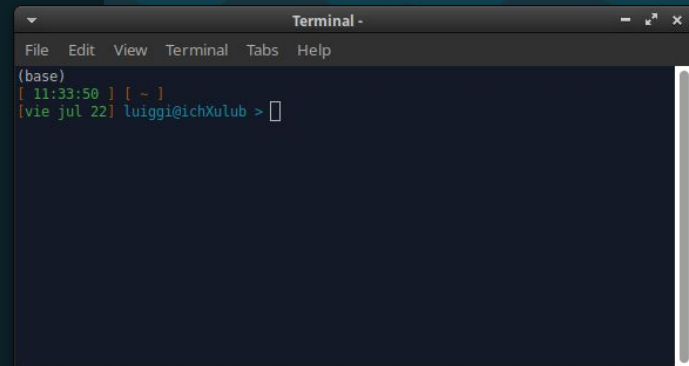
- Puede hacer todo lo que sigue en su computadora local, para ello necesita una terminal de comandos.
 - En el caso de Windows utilice la herramienta llamada **Git bash**, la cual se instala junto con Git y es una ventana de comandos.
 - En MacOS y Linux los comandos se pueden ejecutar en una terminal del sistema.

Git bash



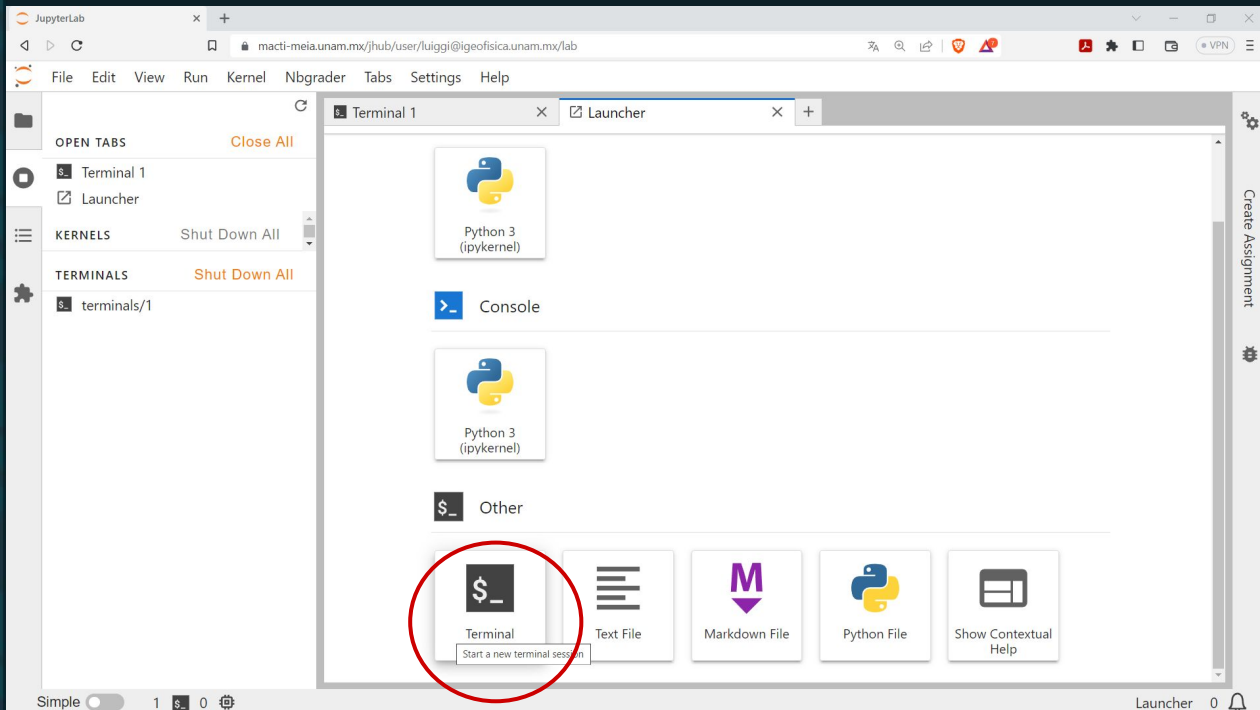
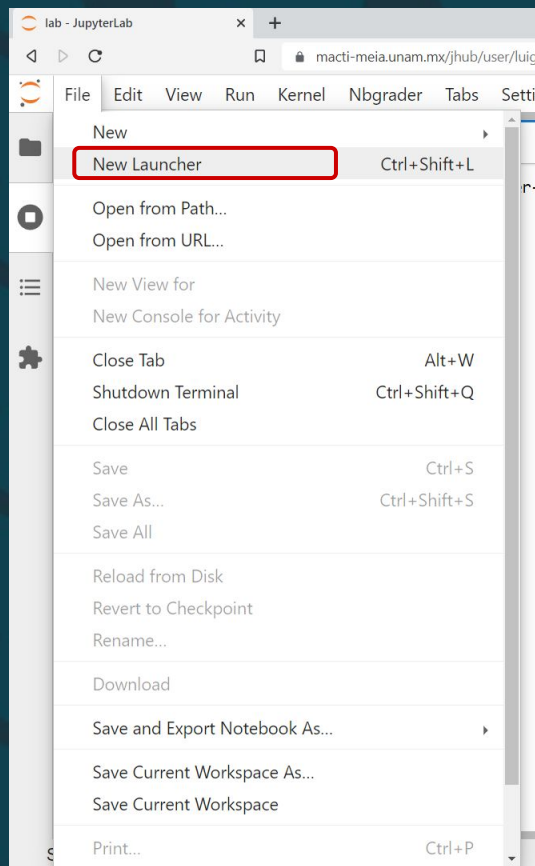
```
MINGW64: c:/Users/luiggi/SuperProyecto
luiggi@MSI MINGW64 ~/SuperProyecto (master)
$
```

Terminal linux



```
Terminal -
File Edit View Terminal Tabs Help
(base)
[ 11:33:50 ] [ ~ ]
[vie jul 22] luiggi@ichXulub >
```

Terminal en Jupyter Lab



Configuración inicial

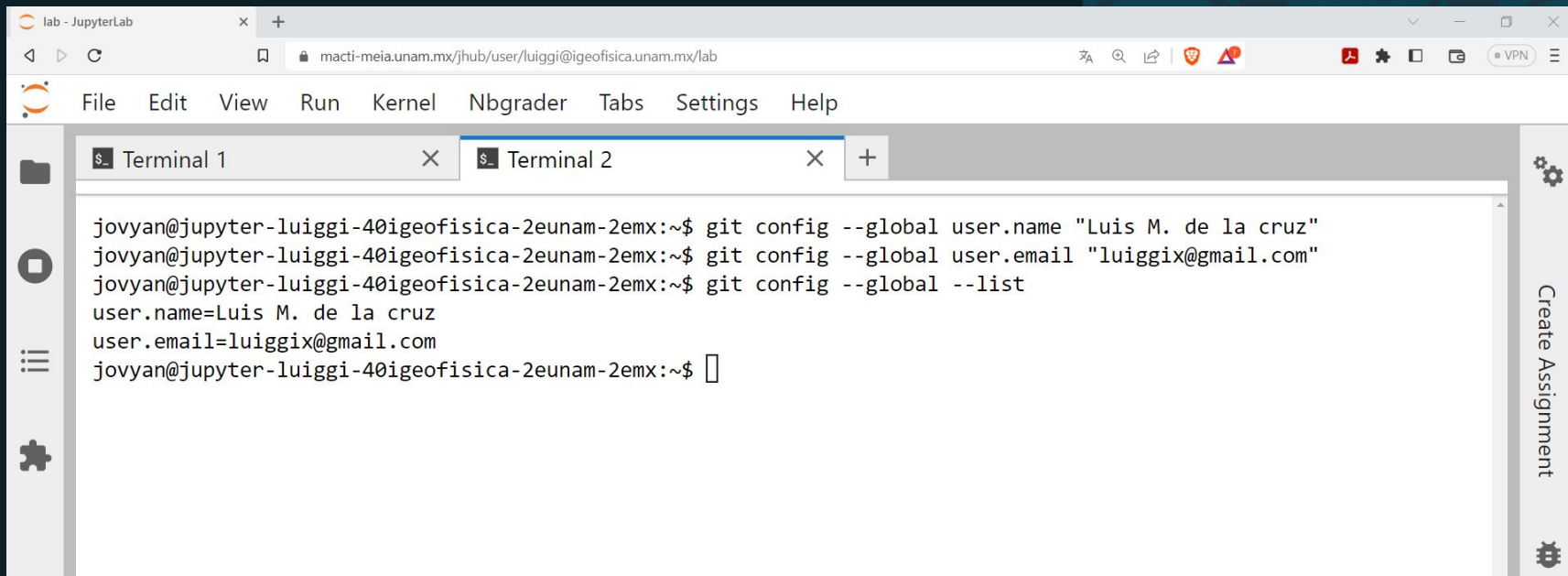
Comenzamos haciendo una configuración inicial. Para que Git identifique el autor de cada cambio en los archivos de un proyecto se debe configurar el nombre de usuario y el correo electrónico. Teclee los siguientes comandos:

```
git config --global user.name "Nombre de usuario"  
git config --global user.email "correo@server.com"  
git config --global --list
```

Escribe un nombre de usuario para identificarte

Escribe tu correo electrónico

Por ejemplo:



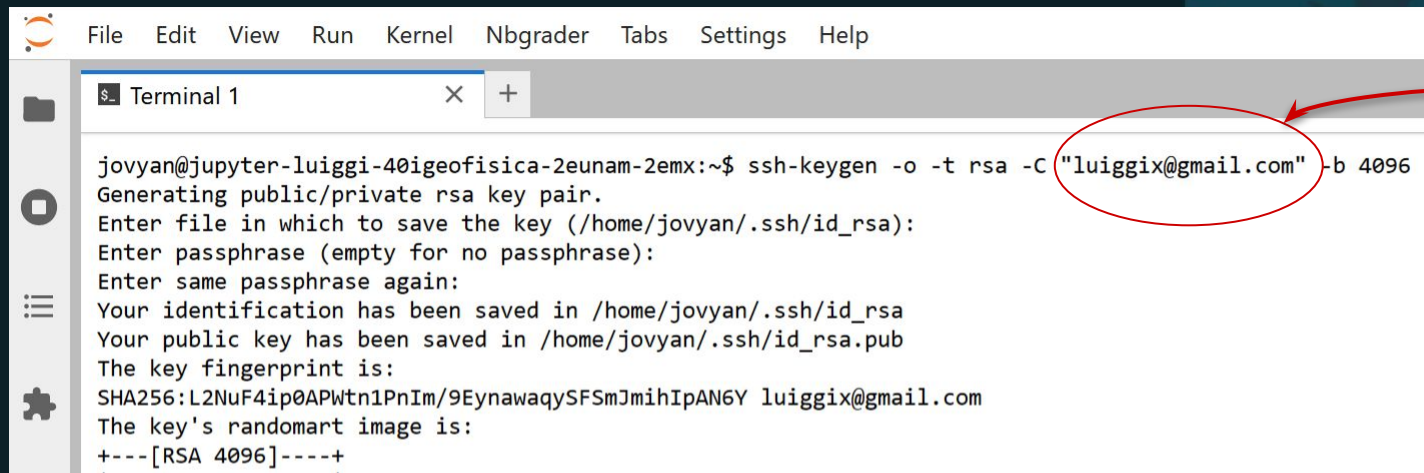
The screenshot shows a JupyterLab web interface. The browser address bar displays the URL `macti-meia.unam.mx/jhub/user/luiggi@igeofisica.unam.mx/lab`. The JupyterLab menu bar includes File, Edit, View, Run, Kernel, Nbgrader, Tabs, Settings, and Help. On the left sidebar, there are icons for a file explorer, a console, and a settings gear. The main area contains two terminal tabs: 'Terminal 1' and 'Terminal 2'. 'Terminal 2' is active and shows the following commands and output:

```
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$ git config --global user.name "Luis M. de la cruz"
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$ git config --global user.email "luiggix@gmail.com"
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$ git config --global --list
user.name=Luis M. de la cruz
user.email=luiggix@gmail.com
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$
```

On the right sidebar, there is a 'Create Assignment' button with a gear icon.

Llave pública y privada

Vamos a usar el protocolo SSH para tener un canal de comunicación segura para compartir información con un repositorio remoto (en este caso con GitHub). Para ello crearemos una llave pública y una privada con el comando `ssh-keygen -o -t rsa -C "usuario@servidor.com" -b 4096`:



```
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$ ssh-keygen -o -t rsa -C "luiggix@gmail.com" -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jovyan/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jovyan/.ssh/id_rsa
Your public key has been saved in /home/jovyan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:L2NuF4ip0APWtn1PnIm/9EynawaqySFSmJmihIpAN6Y luiggix@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
```

Aquí escribes tu correo, con el que te registraste en GitHub

En todas las preguntas solo tecléa <Enter>

Checar las llaves

Para revisar que las llaves se crearon hacemos lo siguiente

```
cat ~/.ssh/id_rsa.pub
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDMryuLMzfAwEMo9mdQB ...  
... UCqQ== "luiggix@gmail.com"
```

```
ls -la ~/.ssh/
```

```
total 20
```

```
drwx----- 2 jupyter...-94051 4096 Jul 20 20:01 .
```

```
drwxr-x--- 9 jupyter...-94051 jupyter...-94051 4096 Jul 20 21:09 ..
```

```
-rw----- 1 jupyter...-94051 jupyter...-94051 3389 Jul 20 20:01 id_rsa
```

```
-rw-r--r-- 1 jupyter...-94051 jupyter...-94051 749 Jul 20 20:01 id_rsa.pub
```

El contenido de la llave pública será usado en GitHub.

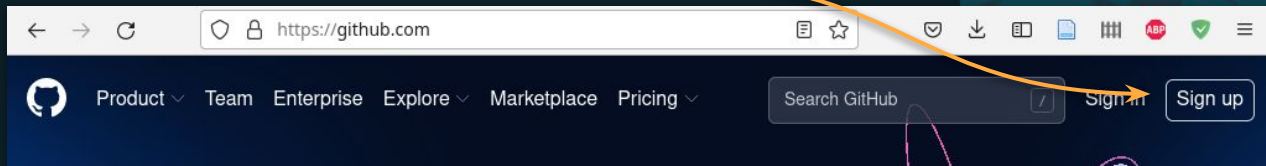


GitHub

Abrir una cuenta.

GitHub es uno de los servicios de alojamiento de proyectos más populares en la actualidad y se basa en Git. Vamos a usar GitHub en combinación con Git, por lo que debes generar una cuenta de usuario:

- Vaya a <https://github.com> y haga clic en **Sign Up**:

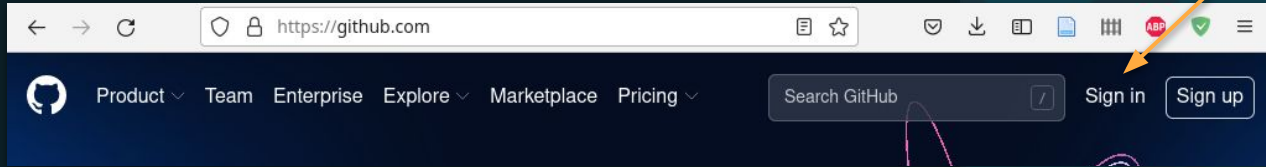


- Siga las instrucciones:

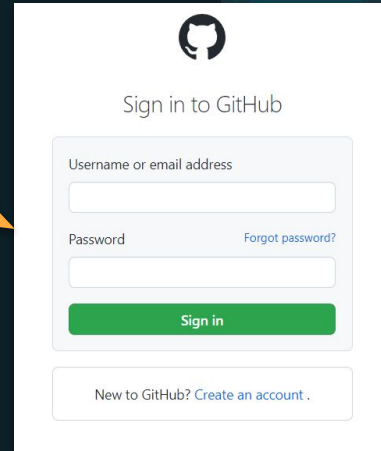
A screenshot of the GitHub 'Welcome to GitHub!' screen. It displays the text 'Welcome to GitHub!' and 'Let's begin the adventure'. Below this, it says 'Enter your email' in teal. There is a text input field with a red arrow pointing to it, and a 'Continue' button to its right.

Ingresa a tu cuenta.

- Una vez creada la cuenta, ingresa a GitHub con tu usuario y contraseña (*Sign in*).



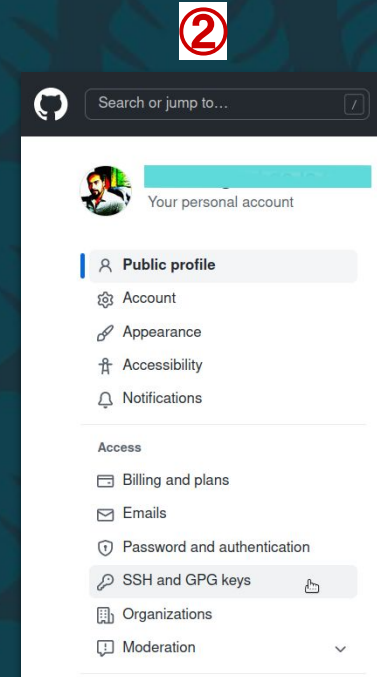
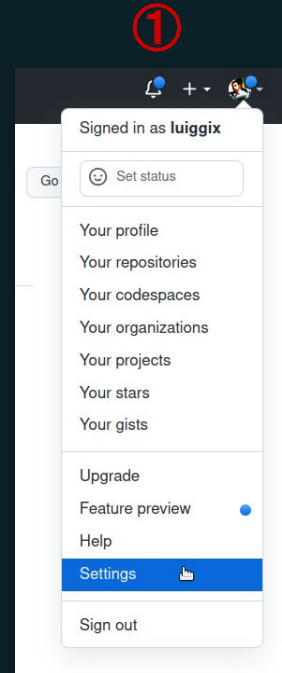
- Escribe tu usuario y contraseña:

A screenshot of the GitHub sign-in form. At the top is the GitHub logo and the text 'Sign in to GitHub'. Below this is a form with two input fields: 'Username or email address' and 'Password'. To the right of the password field is a link that says 'Forgot password?'. Below the input fields is a green button labeled 'Sign in'. At the bottom of the form is a link that says 'New to GitHub? Create an account'. An orange arrow points from the text 'Escribe tu usuario y contraseña:' to the form.

Configuración de llaves SSH

Una vez dentro de tu cuenta en GitHub:

- Ir al menú del perfil personal y hacer clic en **Settings** ① (en la esquina superior derecha)
- Ir a la sección de **SSH and GPG Keys** ② (el menú está a la izquierda)



Configuración de llaves SSH

- Luego hacer clic en **New SSH Key**
- Agregar la llave pública:

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key




Poner un título para identificar la llave

Copiar el texto de la llave pública `id_rsa.pub` aquí.

Finalmente hacer clic en **Add SSH Key**

Para verificar que todo salió bien, teclear el siguiente comando en la terminal:

```
$ ssh -T git@github.com
Hi luiggix! You've successfully
authenticated, but GitHub does
not provide shell access.
```



Creación de un repositorio

Creando un curso

Usando el script: `setup_curso.py`

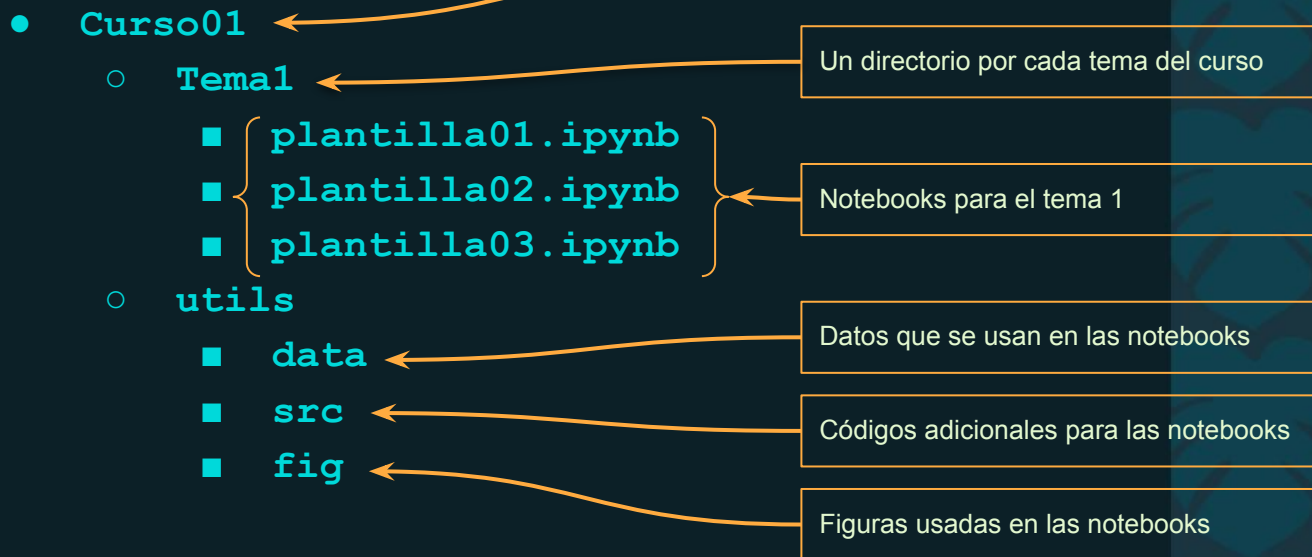
Teclear el nombre del curso.

The screenshot shows the JupyterLab interface with a terminal window open. The terminal displays the command `python setup_curso.py` and its output. The output shows the creation of a course named 'Curso01' and the copying of template files. An orange arrow points from the text 'Teclear el nombre del curso.' to the 'Curso01' entry in the terminal output.

```
jovyan@jupyterter-luiggi-40igeofisica-2eunam-2emx:~$ python setup_curso.py

Nombre del curso : Curso01
Creando el directorio : Curso01/Tema1/
Creando el directorio : Curso01/utils/data/
Creando el directorio : Curso01/utils/src/
Creando el directorio : Curso01/utils/fig/
Copiando plantilla01.ipynb al directorio Curso01/Tema1/
Copiando plantilla02.ipynb al directorio Curso01/Tema1/
Copiando plantilla03.ipynb al directorio Curso01/Tema1/
jovyan@jupyterter-luiggi-40igeofisica-2eunam-2emx:~$
```

Árbol de directorios



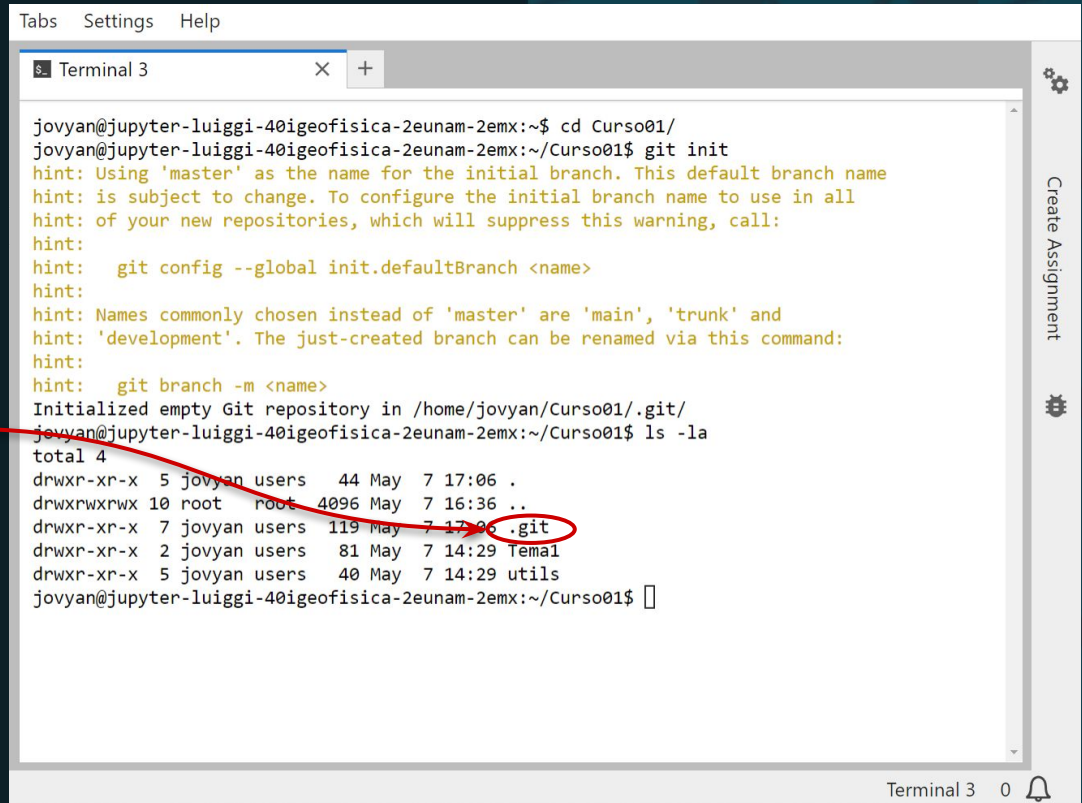
Inicialización del proyecto en Git

```
cd Curso01
```

```
git init
```

```
ls -la
```

Observe que se crea un subdirectorio de nombre **.git** el cual contendrá la base de datos de objetos para monitorear los archivos del proyecto.



```
Terminal 3
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$ cd Curso01/
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/jovyan/Curso01/.git/
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ ls -la
total 4
drwxr-xr-x  5 jovyan users   44 May  7 17:06 .
drwxrwxrwx 10 root   root  4096 May  7 16:36 ..
drwxr-xr-x  7 jovyan users  119 May  7 17:06 .git
drwxr-xr-x  2 jovyan users   81 May  7 14:29 Tema1
drwxr-xr-x  5 jovyan users   40 May  7 14:29 utils
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$
```

Revisar el estado del proyecto y confirmar cambios.

`git status`

En este caso el directorio Tema1/ debe ser añadido al proyecto. Esto se puede realizar usando:

`git add Tema1/`

Observe que el contenido del directorio Tema1/ se agrega al proyecto.

Estos cambios deben ser confirmados:

`git commit -m "Agregué Tema1"`

```
Terminal 3
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Tema1/

nothing added to commit but untracked files present (use "git add" to track)
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git add Tema1/
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git status
On branch master

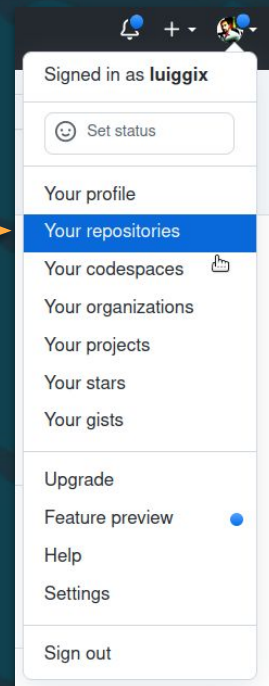
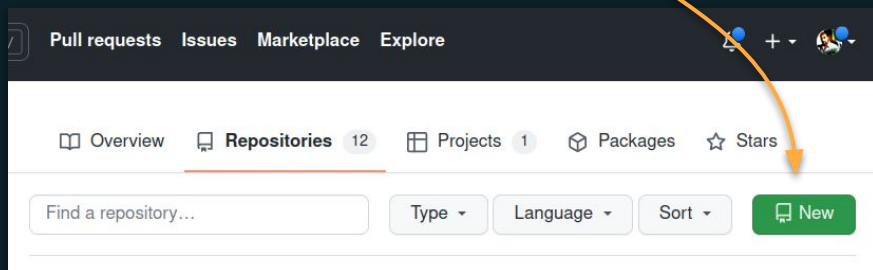
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Tema1/plantilla01.ipynb
    new file:   Tema1/plantilla02.ipynb
    new file:   Tema1/plantilla03.ipynb

jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git commit -m "Agregué Tema1"
[master (root-commit) 9ce4799] Agregué Tema1
 3 files changed, 693 insertions(+)
 create mode 100644 Tema1/plantilla01.ipynb
 create mode 100644 Tema1/plantilla02.ipynb
 create mode 100644 Tema1/plantilla03.ipynb
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$
```

Creación de un nuevo repositorio en GitHub

- Para crear un nuevo repositorio en GitHub debe ir al menú de su perfil y seleccionar *Your repositories*.
- Posteriormente haga clic en el botón *New*.



Creación de un nuevo repositorio


Para este ejemplo solo pondremos un título y una descripción.

Dejamos el proyecto de tipo público e inmediatamente hacemos clic en *Create repository*

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

 luiggix

/

Repository name *

Curso01

✓ Curso01 is available.

Título

Great repository names are short and memorable. Need inspiration? How about [cuddly-adventure?](#)

Description (optional)

Curso en Macti

Descripción

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.


Create repository

22

Creación de un nuevo repositorio

Como el repositorio está vacío, se obtendrá algo como lo que se muestra en la figura:

Quick setup — if you've done this kind of thing before

 Set up in Desktop

 or

HTTPS

SSH

`https://github.com/luiggix/Curso01.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# Curso01" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/luiggix/Curso01.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/luiggix/Curso01.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Sincronizar el proyecto local con el repositorio en GitHub

```
git remote add origin git@github.com:luiggix/Curso01.git
```

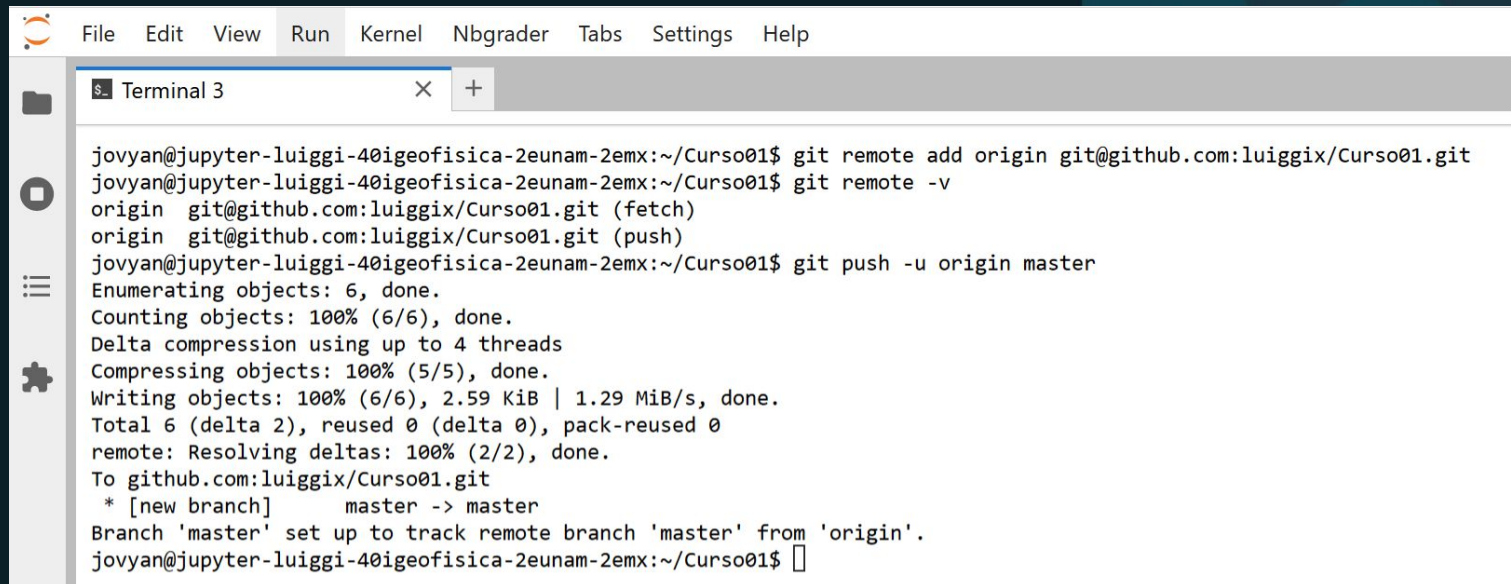
Agregar el repositorio remoto

```
git remote -v
```

Revisar que se haya agregado correctamente

```
git push -u origin master
```

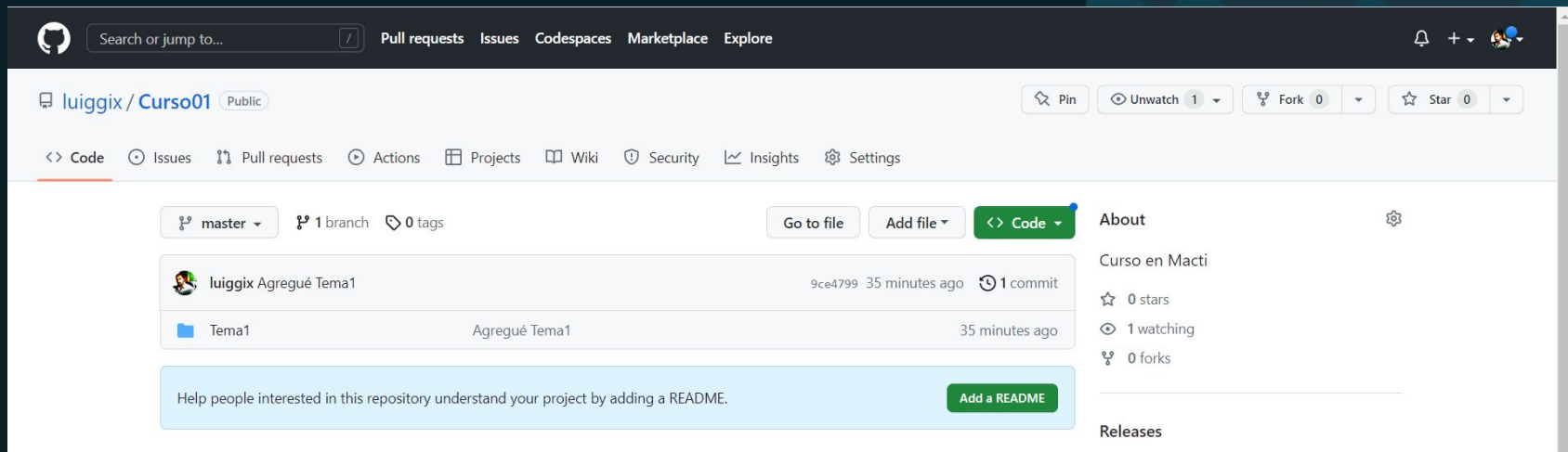
Subir los archivos al repositorio remoto



```
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git remote add origin git@github.com:luiggix/Curso01.git
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git remote -v
origin  git@github.com:luiggix/Curso01.git (fetch)
origin  git@github.com:luiggix/Curso01.git (push)
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 2.59 KiB | 1.29 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:luiggix/Curso01.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Curso01$
```


Repositorio actualizado

Revisamos el repositorio en GitHub y veremos los archivos del proyecto actualizados (no olvide recargar la página para ver los cambios):



The screenshot shows the GitHub interface for the repository 'luiggix / Curso01'. The repository is public and has 0 stars, 1 watch, and 0 forks. The main branch is 'master', and there is 1 branch and 0 tags. The commit history shows a single commit '9ce4799' made 35 minutes ago, titled 'Agregué Tema1'. The repository description is 'Curso en Macti'. There is a button to 'Add a README'.

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

luiggix / Curso01 Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file <> Code

luiggix Agregué Tema1 9ce4799 35 minutes ago 1 commit

Tema1 Agregué Tema1 35 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

Curso en Macti

0 stars

1 watching

0 forks

Releases



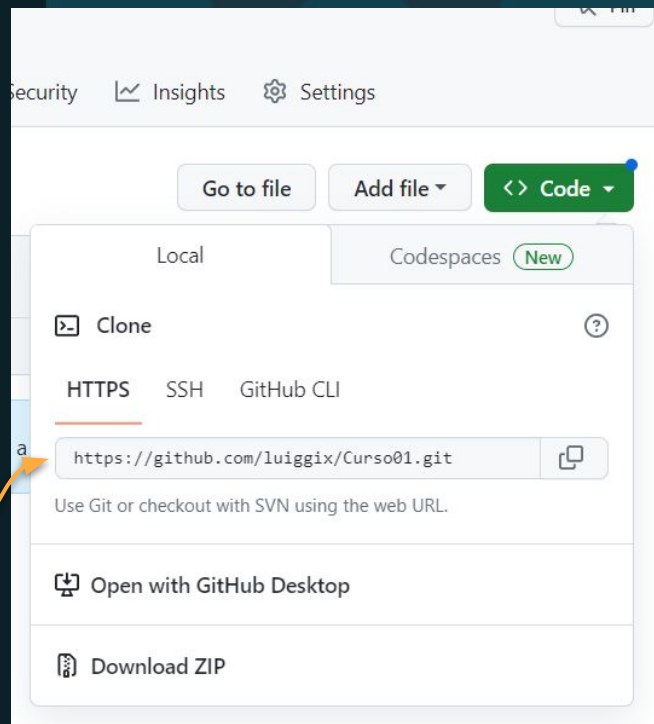
Clonar un repositorio

Git+GitHub. Clonar un proyecto

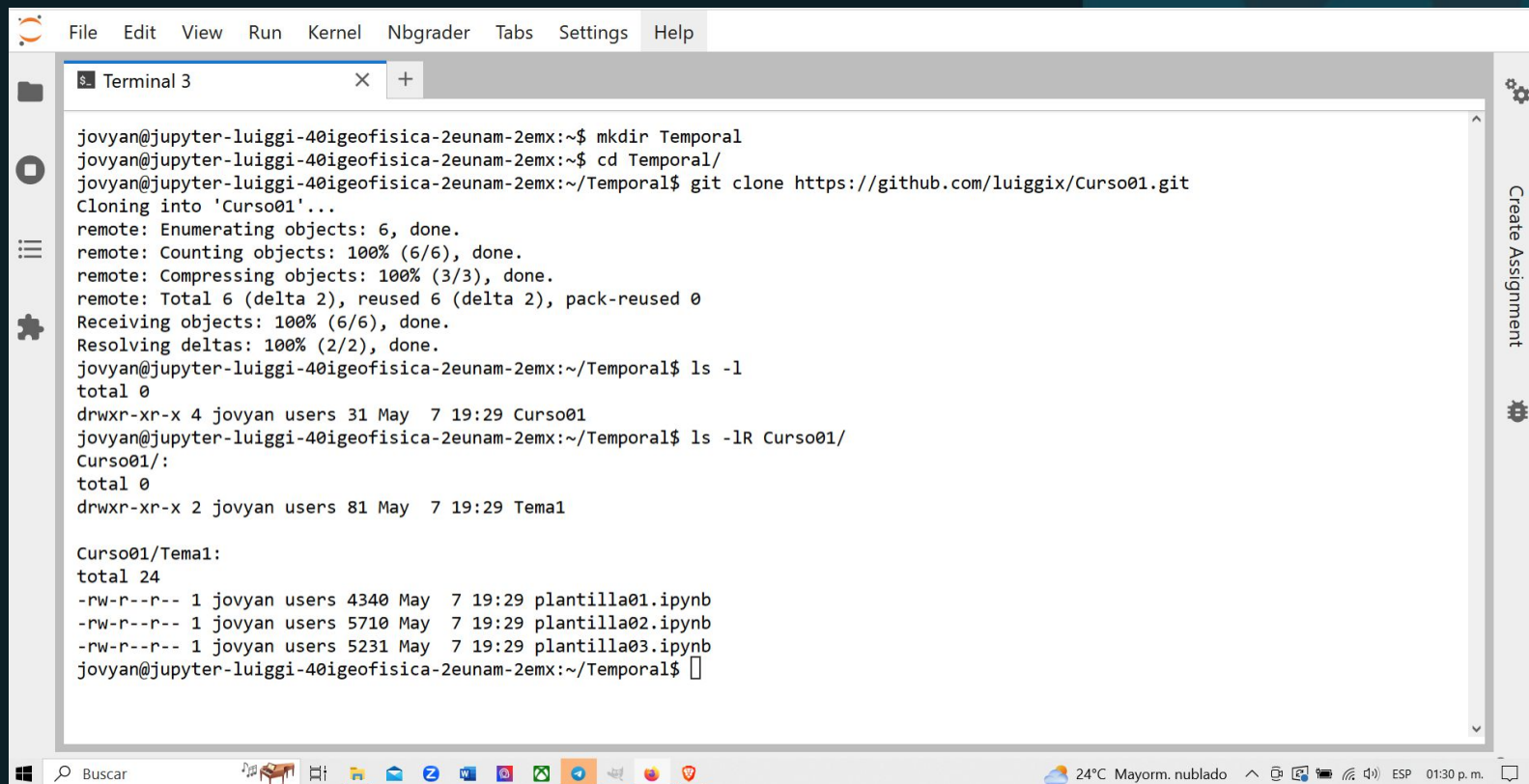
Primero creamos un subdirectorio de nombre **Temporal** y nos cambiamos a ese directorio (esto para evitar modificaciones en el directorio **Curso01**). Entonces los comandos que usamos son:

```
mkdir Temporal  
cd Temporal  
git clone https://github.com/luiggix/Curso01.git
```

La dirección del repositorio la
obtenemos en GitHub



En nuestro ejemplo tenemos:



The screenshot shows a JupyterLab interface with a terminal window titled 'Terminal 3'. The terminal displays the following commands and output:

```
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$ mkdir Temporal
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~$ cd Temporal/
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Temporal$ git clone https://github.com/luiggix/Curso01.git
Cloning into 'Curso01'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 2), reused 6 (delta 2), pack-reused 0
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (2/2), done.
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Temporal$ ls -l
total 0
drwxr-xr-x 4 jovyan users 31 May  7 19:29 Curso01
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Temporal$ ls -lR Curso01/
Curso01/:
total 0
drwxr-xr-x 2 jovyan users 81 May  7 19:29 Tema1

Curso01/Tema1:
total 24
-rw-r--r-- 1 jovyan users 4340 May  7 19:29 plantilla01.ipynb
-rw-r--r-- 1 jovyan users 5710 May  7 19:29 plantilla02.ipynb
-rw-r--r-- 1 jovyan users 5231 May  7 19:29 plantilla03.ipynb
jovyan@jupyter-luiggi-40igeofisica-2eunam-2emx:~/Temporal$
```

The JupyterLab interface includes a menu bar (File, Edit, View, Run, Kernel, Nbgrader, Tabs, Settings, Help), a left sidebar with icons for file explorer, search, and settings, and a right sidebar with a 'Create Assignment' button. The Windows taskbar at the bottom shows the search bar, application icons, and system status (24°C, Mayorm. nublado, 01:30 p.m.).