

Agregando archivos a Github



Universidad de
SanAndrés

Crear un repositorio

En caso de no tener un repositorio, es necesario crearlo. Recordar que es posible tener múltiples repositorios.

- Siguiendo la guía detallada de la página de Github, será posible crear un repositorio.
- Página:

<https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository>

Clonar un repositorio

- Para clonar un repositorio se hará desde la Terminal o Command Prompt, asumiendo que Git se encuentra instalado.
- En segundo lugar, se elige la carpeta donde se desea clonar el repositorio. Recordar que clonar el repositorio es generar una carpeta en la computadora personal con los archivos del repositorio y una carpeta adicional `.git` que se encuentra oculta para el *File Explorer* o *Finder*.

Para ellos en Linux o MacOS ejecutando el siguiente comando es posible saber en qué lugar nos encontramos.

Linux

```
$ pwd  
/home/udesá
```

MacOS

```
$ pwd  
/Users/udesá
```

Clonar un repositorio

- En Windows, el siguiente comando indica en qué lugar nos encontramos.

Windows

```
$ cd  
C:/Users/udesa
```

Clonar un repositorio

- Recordar que si existe una carpeta dentro de la carpeta udesa, es posible ingresar y también salir de ella.

Linux / MacOS

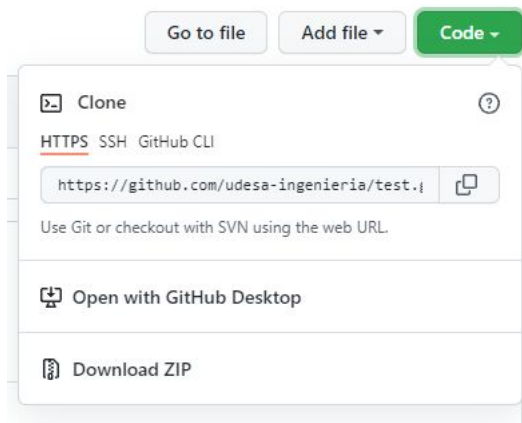
```
$ cd pensamiento_computacional
$ pwd
/Users/udesa/pensamiento_computacional
```

Windows

```
$ cd pensamiento_computacional
$ cd
C:/Users/udesa/pensamiento_computacional
```

Clonar un repositorio

Una vez en la carpeta o directorio que se desea clonar el repositorio, obtener la dirección del repositorio desde:



- Es importante que el link comience con <https://>.
- Una vez copiado el link, ejecutar en la terminal el comando, pegando el link como en el siguiente ejemplo.:

```
$ git clone https://github.com/udes-a-ingenieria/test.git
```

- Revisar que en la carpeta donde clonaron, se encuentre la carpeta del repositorio, y sus archivos que tenga que tener.

Verificar Cambios

- Dentro del repositorio **local**, crear un archivo `changes.py`, como ejemplo. Inventar un código de Python que funcione a modo de ejemplo nuevamente.
- Para revisar si existen cambios y ver las diferencias en el repositorio **local**, ejecutar:

```
$ git status
```

```
$ git diff
```

- En rojo aparecen los cambios, y para seleccionar para agregarlos al repositorio es posible hacerlo de tres formas distintas.

```
$ git add <file-name>
```

```
$ git add .
```

```
$ git add -A
```

- Aparecerá en verde, seleccionado para **crear un commit** y hacer push al **repositorio remoto**.

Crear un commit

- Una vez seleccionado los archivos candidatos para crear un commit, es decir, que se encuentren en color verde luego de hacer `git status`. Se creará un commit con el siguiente comando:

```
$ git commit -m "Mensaje Descriptivo de los cambios"
```

- Para verificar que el commit fue realizado correctamente ejecutar:

```
$ git log
```

- Debe aparecer el commit con el mensaje que agregaron en el paso anterior.

Push

- Creado el commit, sólo queda hacer un push. Probar con uno de los dos comandos:

```
$ git push
```

```
$ git push origin main
```

- Revisar que el mensaje diga que está todo ok. Luego, ir a Github y revisar que el archivo `changes.py` está subido.

IMPORTANTE: Si Git pide usuario y contraseña para hacer un push, continuar con el paso de autenticación con Github CLI.

Autenticación

Para manejar repositorios, es necesario estar autenticados con la cuenta que tiene acceso a los repositorios.

La mejor forma es utilizar [Github CLI](#). Una vez instalado es necesario realizar los siguientes pasos:

1. Generar un Personal Access Token (PAT).
2. Autenticarse con Github CLI.

Personal Access Token

La generación del access token se realiza en la cuenta de Github.

1. Dirigirse a *Profile > Settings*.
2. Ingresar en *"Developer Settings"*.
3. Luego, en *Personal Access Tokens*, tocar el boton *Generate New Token*.
4. Elegir, *Expiration > No Expiration*.
5. Añadir una nota, y los permisos *'repo'*, *'read:org'*, *'workflow'*.
6. Guardar el token generado en algún lugar seguro (no se puede volver a ver en Github).

Personal access tokens

[Generate new token](#)[Revoke all](#)

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_P2Y639sQ2lJNXF8E0SsOesGsvqfaex1wNw0h 

[Delete](#)

Github CLI

Para autenticarse con Github CLI:

1. Abrir el Command Prompt de Windows o Terminal en Linux/macOS.
2. Ejecutar el comando:

```
$ gh auth login
```

3. Presionar **enter** hasta que aparezca la pregunta: *"How would you like to authenticate Github CLI?"*. Elegir *"Paste an authentication token"*.
4. Pegar el token generado y presionar **enter**.