

Clase 0
Fisica Computacional 2
Jhovanny Andres Mejia Guisao

A continuacion se presentan algunos link utiles para manejar **git** ot **github**

Git Tutorial

<https://web.archive.org/web/20161121145226/http://rypress.com:80/tutorials/git/index>

<https://help.github.com/en/github/getting-started-with-github/set-up-git>

Github

<https://help.github.com/en/github/getting-started-with-github/create-a-repo>

<https://guides.github.com/activities/hello-world/>

Obtener cambios de un repositorio remoto

<https://help.github.com/es/github/using-git/getting-changes-from-a-remote-repository#further-reading>

Git-workflow or feature branching

<https://gist.github.com/blackfalcon/8428401>

Empecemos con un breve repaso

Como clonar y trabajar en un repositorio:

<https://github.com/>

(cree una cuenta si aun no tiene)

Muévase al repositorio del curso.

https://github.com/jmejiagu/CursoFCII-UdeA_20231

Haga un “Fork”

The screenshot shows a GitHub repository page for 'jmejiagu / CursoFCII-UdeA_20231'. The repository is public and has 0 stars, 0 forks, and 1 watcher. The main branch is 'main'. The repository contains a folder 'lectures/clase0' and files '.gitignore' and 'README.md'. The 'README.md' file is open, showing the title 'CursoFCII-UdeA_20231' and a description: 'Repositorio con códigos e información acerca de la materia Física Computacional 2 en el instituto de física de la UdeA'. The right sidebar shows the 'About' section with the repository description, 'Releases' section with 'No releases published', and 'Packages' section with 'No packages published'.

Usted puede hacer (casi)todo online, o también puede clonar el repositorio y trabajar "en la computadora" (Lo ultimo es lo recomendado)

```
git clone git@github.com:jmejiagu/CursoFCII-UdeA_20231.git
```

Una vez que tenga **SU** repositorio, puede trabajar en el (recuerde que puede ser online también):

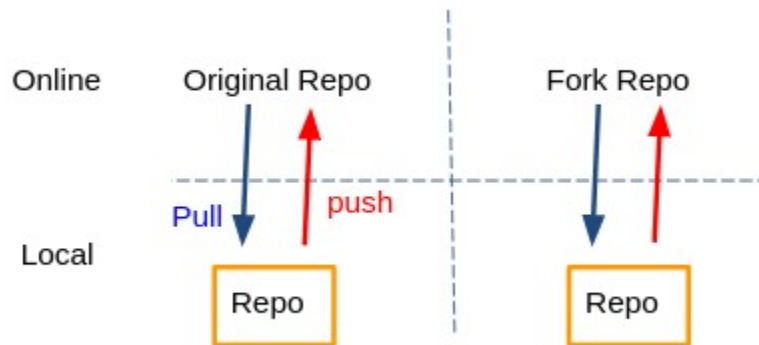
```
git status
cree, edite y guarde un archivo de prueba (emacs test.txt &)
git add test.txt
git commit -m "add txt test file"
git push (para esto, ya debe haber establecido su clave ssh. Ver al
final de este texto).
```

Ahora, borremos ese mismo archivo

```
git rm test.txt
git status
git commit -m "delete txt test file"
git push
```

En algun momento es posible que "git" le pida que se identifique:

```
git config --global user.name "Your Name"
git config --global user.email your.email@example.com
```



NOTA BENE. si usted esta trabajando online entonces puede actualizar su repositorio local de la siguiente manera:

```
 cree un archivo de prueba online
git fetch (git fetch remotename)
git merge (remotename/branchname)
```

`git pull` es un atajo conveniente para realizar tanto `git fetch` y `git merge` en el mismo comando:
`git pull` (git pull remotename branchname)

Asi puede ver todos sus commits (cambios):

```
git log --oneline (git log)
```

TENGA EN CUENTA

Para sincronizar un repositorio producto de un fork con respecto al repositorio original: Solo es necesario hacerlo una vez en la copia local del repositorio

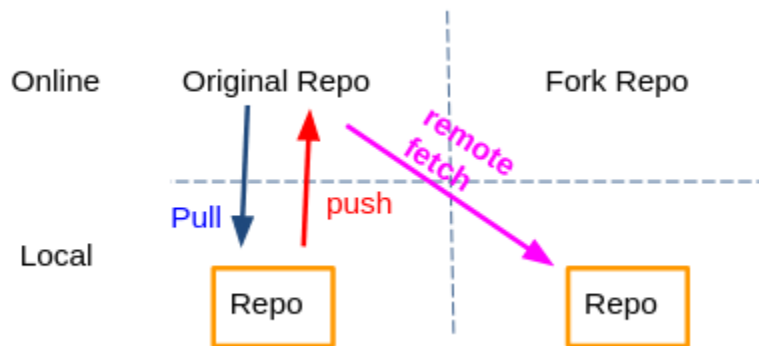
<https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/configuring-a-remote-for-a-fork>

- 1) `git remote -v`
- 2) `git remote add upstream git@github.com:jmejiagu/CursoFCII-UdeA_20231.git`
- 3) `git remote -v`

acciones necesarias cada vez que haya que hacer la sincronización:

<https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/syncing-a-fork>

```
git fetch upstream
git checkout main (Esto si usted no esta en la rama main)
git merge upstream/main
```



nota sobre ssh key:

Valla a su archivo .ssh (en la computadora que usted este) y mire la clave publica que ahi tiene.
Entoces, ponga esa clave en cuenta de git (su perfil en internet)

Seeting - > ssh keys (copie y pegue la clave de su .ssh aca)

NOTE: tenga cuidado como abre su archivo (.ssh/id_rsa.pub) para que la clave se copie bien sin espacios (mejor use la version de emacs que no es en la consola)