flujo drive github

**ORDEN**:

* **git init** o **git clone** (para crear o clonar un repositorio y tener la carpeta oculta .git)

→ normalmente los haremos dentro de la carpeta “ironhack”: **git clone + <repository link>**

* **git status** (para ver si realmente nuestra carpeta es un repositorio)
* **git checkout - b <branch name>** (para cada assignment de clase crearemos una nueva rama)
  + git checkout  **<branch name>** (para entrar a una rama sin querer crearla)
* *// creamos archivos, trabajamos en ellos*
* **git add <filename>** (para añadir el nuevo archivo o los cambios en el archivo a la nueva rama *- se añade a staging -*)

\* **git add .** → añade **todos** los archivos (es buena practica hacerla yendo a la base de la ruta)

\* **git restore** **<filename>** (si queremos descartar este cambio)

* **git commit -m “<mensaje>”**
  + **git commit -m "lab-started"** (mientras trabajamos)
  + **git commit -m "lab-finished"** (cuando acabamos de trabajar)
* **git checkout main** (o git switch main) [antes de hacer el merge de la rama nueva a la main, deberemos de entrar en la main]
* **git merge <nombre de la nueva rama>** (estaremos copiando todo a la main)
* **git push +** [**“remote link” (con el token**](https://docs.google.com/document/d/1FRFjLHP06Q-SMj-qcDKxeFNR55OWlmNHy2nD6qWBXCI/edit)) (si ponemos el remote link hacemos un push de TODO) o **git push origin <branch name** (intentar que sea la main)**>** (si el repositorio ya está clonado)

***\* ANTES DE HACER EL PUSH HAY QUE hacer un git pull o git fetch →* git pull origin main***(en el caso de que haya habido cambios en el repositorio remoto. Un git pull hace una “fusión” con nuestro main, en cambio un fetch te copia el repositorio remoto en un nuevo branch local, por lo que luego tendremos que hacer un merge). También podemos* **git pull +** [**“remote link” (con el token**](https://docs.google.com/document/d/1FRFjLHP06Q-SMj-qcDKxeFNR55OWlmNHy2nD6qWBXCI/edit)).

* **pull request dentro de la interfaz de git hub →** “create pull request”

IMPORTANTE: cada vez que hagamos algo nuevo, crearemos una RAMA nueva de nuestro MAIN, después haremos un GIT ADD y un GIT COMMIT, iremos a la rama MAIN y haremos un merge de la nueva a la main

FLUJO DE TRABAJO ENTRE PLATAFORMAS

1 - **Al final del curso** copiar **de my repo local** **a G.Drive** como carpetas principales:

* dataptmad1121\_labs
* dataptmad1121\_lessons

2 - todos los **ejercicios o challenges (slack, katas)** intentar hacerlo en la carpeta de **ejercicios\_drive\_local\_remoto** para luego pasarlos a sus respectivas carpetas:

* *ejercicios\_drive\_local\_remoto*
* **G.Drive**
* **local**
* **remoto**

*\* Si se hacen desde otra carpeta como dataptmad1121\_labs seguir el siguiente flujo:*

* **local:** challenges - katas
* **G.Drive**: challenges - katas
* **remoto**: challenges - katas y dataptmad1121\_labs

3 - Los **labs** hacerlos en sus respectivas carpetas, pero luego pasarlos también a:

* **local:** challeges - labs
* **G.Drive**: challenges - labs
* **remoto**: challenges - labs y dataptmad1121\_labs

4 - toda la **teoría** hacerla primero **en G.Drive** para pasarla luego a **local** - **remoto**

5 - Si en un repo de los **profes añaden algo nuevo**:

1. abrir ese archivo en su repo, modificarlo añadiendo un espacio
2. hacer un fetch de su repo al mio (por ahora todo remoto en UI)
3. hacer un push de mi repo local al remoto para que no haya nada pendiente
4. hacer un pull con ese nuevo archivo de los profes

→ RESULTADO el repo local se habrá mezclado sin sobre escribirse

1. subir el archivo a drive

6 - si **para un lab no he creado una nueva rama**:

1. abrir el archivo y modificar algo mínimo
2. crear una nueva rama
3. hacer un merge de la rama main a esta nueva
4. git add .
5. git -m “...”
6. git push origin >nuevo nombre de la rama>