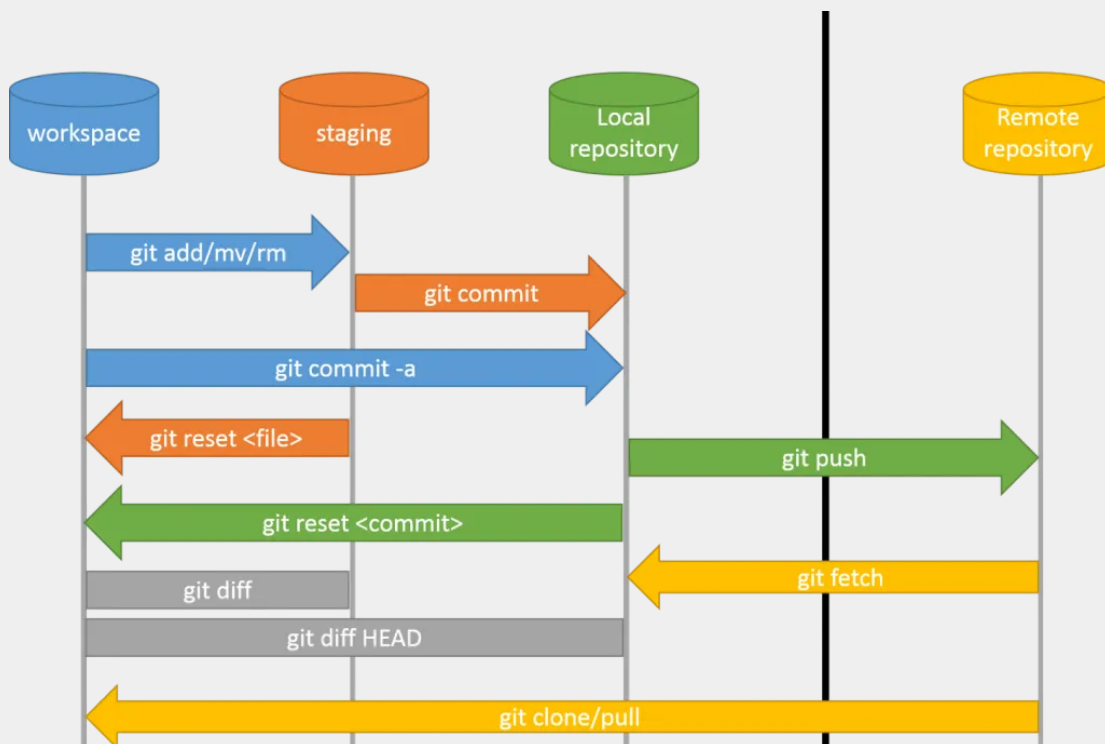


Activitat 15: Creació repositori local i remot

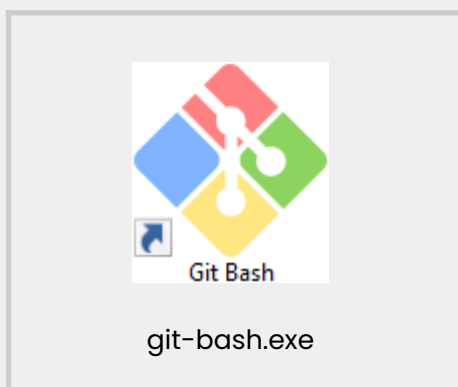
Imatge obtinguda de blog.podrezo.com



Abans de començar, cal llegir el que us demano al final del document.

Confirmar que l'aplicació `git` està instal·lada.

Obrir l'aplicació `git-bash` que teniu instal·lat al vostre ordinador.



Comprovar que teniu **git** instal·lat i instal·lat correctament, podem comprovar quina versió de **git** tenim instal·lada.

```
root@Win2354 MINGW64 /d/  
$ git --version  
git version 2.27.0.windows.1
```

Si no el teniu instal·lat, podeu obtenir el programa instal·lable a l'[enllaç](#). Cal descarregar la versió que es troba sota l'[enllaç de Windows](#) i a la secció **Standalone Installer**, i d'aquí la versió **64-bit Git for Windows Setup**.

Configuració

En principi tots ja teniu ben configurat el vostre ordinador.

Configuració del nom d'usuari

Per conèixer el **nom d'usuari**, que està configurat, cal fer servir la següent ordre:

```
root@Win2354 MINGW64 /d/  
$ git config --global user.name  
joanpardogine
```

I aquesta ens tornarà el **nom d'usuari**, que està configurat actualment al nostre ordinador.

Configuració de l'adreça de correu de l'usuari

Per conèixer l'**adreça de correu electrònic**, que està configurat, cal executar la següent comanda:

```
root@Win2354 MINGW64 /d/  
$ git config --global user.email  
joanpardo@ginebro.cat
```

I aquesta ens tornarà l'**adreça de correu electrònic**, que està configurat actualment al nostre ordinador.

Per tant, només caldrà que el segueixin aquells que pel motiu que sigui no el tenen ben configurat.

Passos per configurar el nom d'usuari

El primer que cal confirmar que teniu ben configurat el **git** que farem servir. Els paràmetres que són imprescindibles són el **nom d'usuari** i l'**adreça de correu electrònic**, ja que aquesta informació quedarà registrada a totes les actualitzacions que fem a **git**. Aquesta informació ha de coincidir amb la de l'usuari que tingueu creat a **github.com**.

Per configurar el **nom d'usuari**, farem servir la següent ordre:

Sintaxi de l'ordre `git config`

```
git config --global user.name <nomUsuariGithub>
```

A on **<nomUsuariGithub>** és el nom d'usuari del vostre compte de **github.com**.

Per exemple, el meu compte de **github.com** té com a usuari **joanpardogine**. Per tant, en el meu cas l'ordre a executar seria:

```
root@Win2354 MINGW64 /d/  
$ git config --global user.name "joanpardogine"
```

Passos per configurar l'adreça de correu electrònic

Per configurar l'**adreça de correu electrònic**, farem servir el mateix procediment que hem fet servir per configurar el **nom d'usuari**, però en aquest cas farem servir la següent ordre:

```
git config --global user.email <adreçaDeCorreuElectrònic>
```

A on `<adreçaDeCorreuElectrònic>` és l'**adreça de correu electrònic** del vostre compte de `github.com`.

Per exemple, el meu compte de `github.com` té com a adreça de correu electrònic `joanpardo@ginebro.cat`. Per tant, en el meu cas l'ordre a executar seria:

```
root@Win2354 MINGW64 /d/  
$ git config --global user.email "joanpardo@ginebro.cat"
```

Ara que ja tenim l'entorn configurat, procedirem a crear el nostre repositori des de zero.

Creació de la carpeta per aquesta activitat

Començarem creant una carpeta a on guardarem tots els nostres repositoris. Crear una carpeta `SMX2-2022-2023`

```
root@Win2354 MINGW64 /d/  
$ mkdir SMX2-2022-2023
```

Entrar a la carpeta `SMX2-2022-2023`

```
root@Win2354 MINGW64 /d/SMX2-2022-2023
$ cd SMX2-2022-2023
```

Crear una carpeta **MP07**,

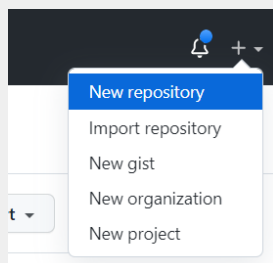
```
root@Win2354 MINGW64 /d/SMX2-2022-2023
$ mkdir MP07
```

Entrar a la carpeta **MP07**.

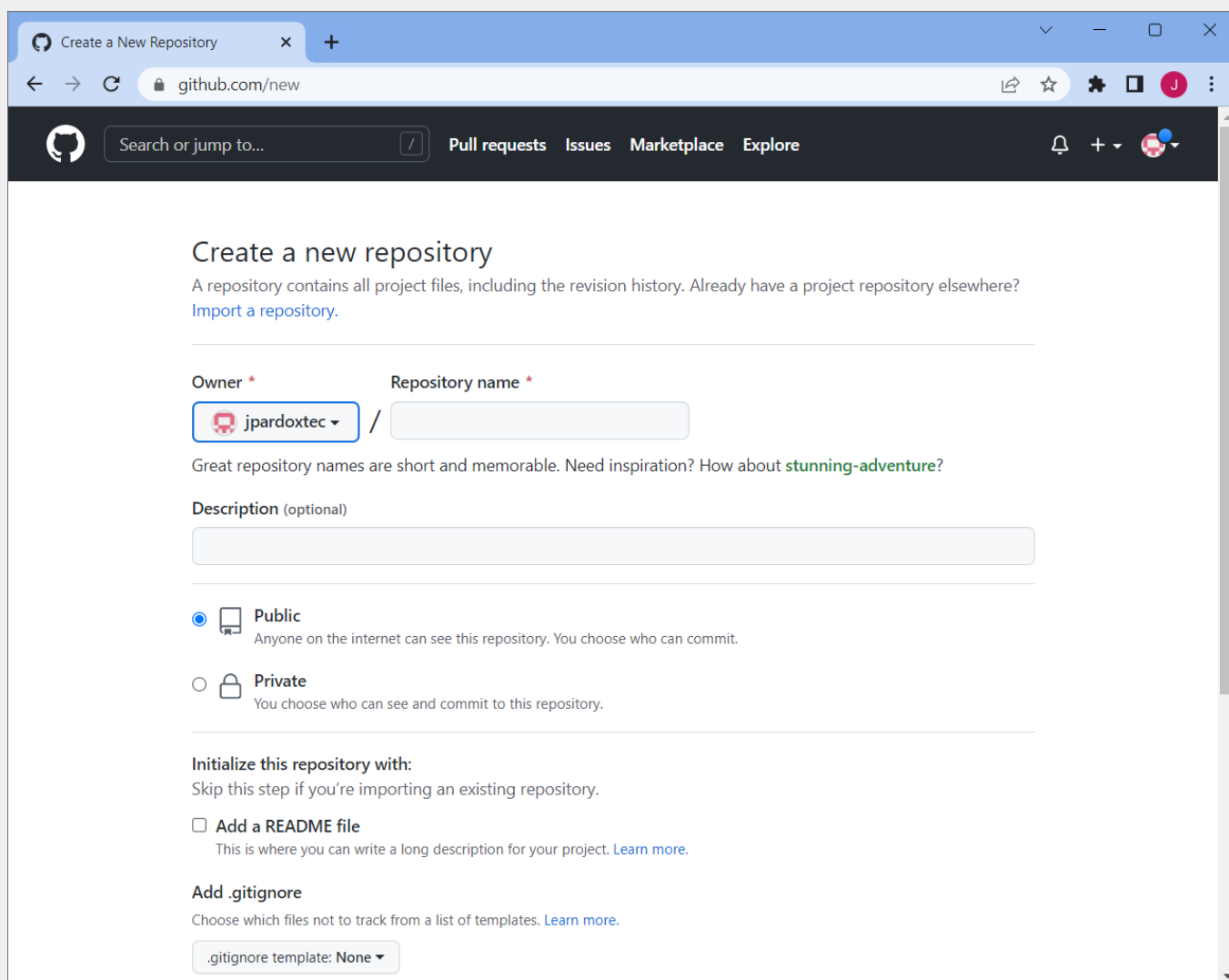
```
root@Win2354 MINGW64 /d/SMX2-2022-2023
$ cd MP07
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07
$
```

Creació del vostre repositori remot (al github.com)

Entrar al compte al github.com. A la part superior dreta apareixerà l'opció per crear un nou repositori (**New repository**) al desplegable del signe **+**.



En pitjar aquesta opció s'obre la finestra per **crear un nou repositori** (**Create a new repository**).



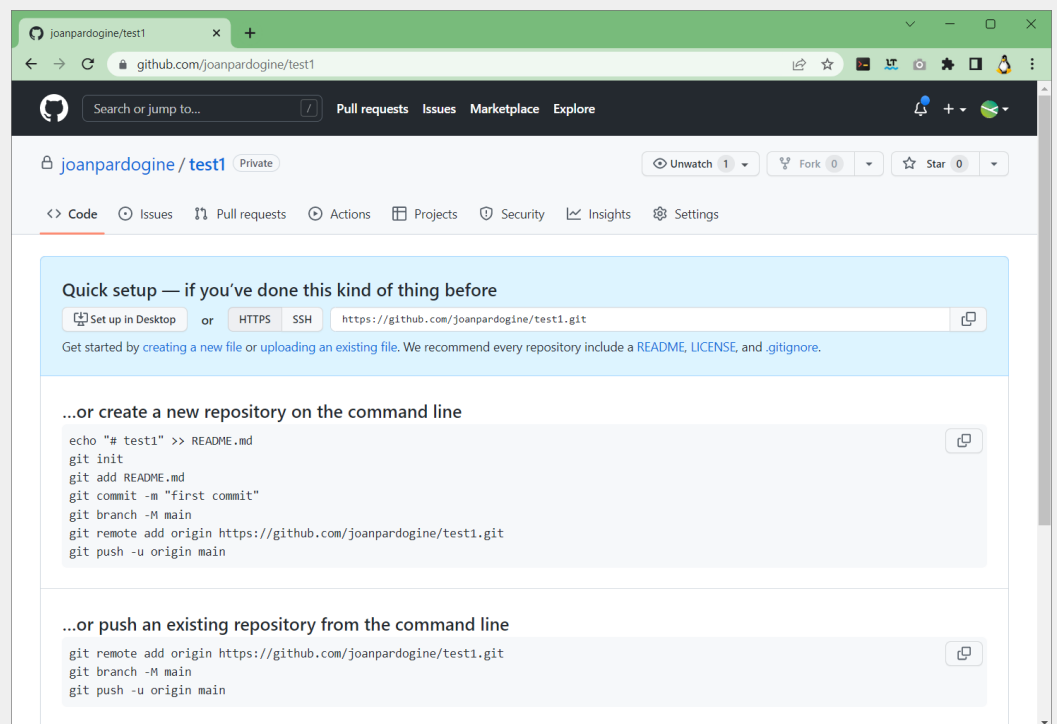
En el nostre cas, al camp **nom del repositori (Repository name)**, que com indica l'asterisc és obligatori, escriurem **<CognomALumne>-A015U** (important respectar tant els guions com les majúscules o minúscules del nom del repositori.), us recordo que a on diu **<CognomALumne>** vol dir que cal escriure el **cognom de l'alumne**, sense el **nom de l'alumne**.

També seleccionarem l'opció de **privat (Private)** i pitjar el botó **Create repository** per crear el nou repositori remot.

I per últim, però no per això menys important, cal que convideu a l'usuari **joanpardogine** com a col·laborador.

Detalls del repositori	
Nom:	<CognomAlumne>-A015U
Tipus:	Privat
Colaborador:	<i>joanpardogine</i>

A continuació, el propi **github**, ens ofereix un seguit de comandes per ajudar-nos a crear el nostre repositori local i enllaçar-lo amb el repositori remot.



```
echo "# <CognomAlumne>-A015U" >> README.md
```

```
git init (1)
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main (2)
```

```
git remote add origin
```

```
https://github.com/<usuariGitHub>/<CognomAlumne>-A015U.git
```

```
git push -u origin main
```

(1) aquesta comanda només cal que l'executeu si no l'heu executat anteriorment, ja que si l'executeu en un repositori ja inicialitzat, perdeu tots els canvis que tingueu fets, pel fet que inicialitza el repositori, és a dir crea un repositori buit.

(2) aquesta comanda només cal que la facin aquells alumnes que tenen la branca com a **master** el compte de **main**.

Creació del vostre repositori local

El següent pas serà crear el nostre **repositori local**, tornarem allà a on ho hem deixat al darrer punt de [Creació de la carpeta per aquesta activitat](#).

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07
$ pwd
/d/SMX2-2022-2023/MP07
```

Un cop ja ens trobem situats a la carpeta que ens interessa

(`D:\SMX2-2022-2023\MP07`) crearem una carpeta per ubicar el nostre repositori local, amb el nom `<CognomAlumne>-A015U` (Important respectar tant els guions com les majúscules o minúscules del nom del repositori.), us recordo que a on diu `<CognomAlumne>` vol dir que cal escriure el **cognom de l'alumne**, sense el **nom de l'alumne**.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07
$ mkdir <CognomAlumne>-A015U
```

Entrar a la carpeta `<CognomAlumne>-A015U`


```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07
$ cd <CognomAlumne>-A015U
```

Abans de res cal crear algun fitxer dins del nostre **repositori local**. Per fer-ho executarem la comanda `echo "# <CognomAlumne>-A015U" >>`

`README.md`. Aquesta comanda el que fa és afegir el text `#`

`<CognomAlumne>-A015U` dins del fitxer `README.md`. I si aquest fitxer no existeix el crea.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
$ echo "# <CognomAlumne>-A015U" >> README.md
```

Per veure que hi ha dins d'aquesta carpeta executem `ls -la`.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
$ ls -la
total 17
drwxr-xr-x 1 root root 0 Sep 19 16:04 ./
drwxr-xr-x 1 root root 0 Sep 19 16:03 ../
-rw-r--r-- 1 root root 7 Sep 19 16:04 README.md
```

A continuació Inicialitzarem el repositori amb `git` dins d'aquesta carpeta.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
$ git init
Initialized empty Git repository in
D:/SMX2-2022-2023/MP07/<CognomAlumne>-A015U/.git/
```

Sintaxi de l'ordre `git init`

```
git init [-b <nomBranca> | --initial-branch=<nomBranca>]
[<nomDirector>]
```

Opcions

-b <nomBranca> --initial-branch=<nomBranca>	Utilitza el nom especificat (<nomBranca>) per a la branca inicial del repositori acabat de crear. Si no s'especifica, anomenarà main (valor per defecte) a la branca inicial del repositori.
<nomDirectori>	Si proporcioneu el nom d'un directori (<nomDirectori>), l'ordre s'executarà al seu interior. I si aquest directori no existeix, es crearà.

Comprovem en quin estat es troba el repositori.

```

root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ git status
On branch main

No commits yet

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

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

Sintaxi de l'ordre git status

```
git status [<opcions>...] [--] [<nomFitxer>...]
```

Per veure que hi ha dins d'aquesta carpeta executem **ls -la**.

```

root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ ls -la
total 5
drwxr-xr-x 1 root root  0 Sep 19 18:05 ./
drwxr-xr-x 1 root root  0 Sep 19 16:23 ../
  
```

```
drwxr-xr-x 1 root root  0 Sep 19 18:05 .git/
-rw-r--r-- 1 root root 18 Sep 19 16:58 README.md
```

I veiem que hi ha una carpeta que ha creat **git** per guardar tota la informació de l'estat i dels canvis del nostre repositori. El nom d'aquesta carpeta és **.git** i està oculta.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ ls -la .git
total 11
drwxr-xr-x 1 root root  0 Sep 19 18:05 ./
drwxr-xr-x 1 root root  0 Sep 19 18:05 ../
-rw-r--r-- 1 root root 21 Sep 19 18:05 HEAD
-rw-r--r-- 1 root root 130 Sep 19 18:05 config
-rw-r--r-- 1 root root 73 Sep 19 18:05 description
drwxr-xr-x 1 root root  0 Sep 19 18:05 hooks/
drwxr-xr-x 1 root root  0 Sep 19 18:05 info/
drwxr-xr-x 1 root root  0 Sep 19 18:05 objects/
drwxr-xr-x 1 root root  0 Sep 19 18:05 refs/
```

Els fitxers que farem servir a aquesta activitat, i els podeu trobar a:

- [hora_01.html](#) i [hora_01.js](#)
- [hora_02.html](#) i [hora_02.js](#)
- [hora_03.html](#) i [hora_03.js](#)

Ara procedirem a crear un fitxer anomenat **hora_01.html**. Podem crear el fitxer o simplement copiar-lo a la carpeta a on hem iniciat el nostre repositori.

Ho podeu fer amb el vostre editor. En aquest exemple l'obrirem amb l'**IDE Visual Studio Code**.

```
root@Win2354 MINGW64
/d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
```

```
$ code hora_01.html
```

git

```
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta charset="utf-8">
    <title>Temps (hores, minuts i segons).</title>
  </head>
  <body>
    <p id="rellotge"></p>
    <p id="informacio"></p>
    <script type="text/javascript"
      src="./hora_01.js"></script>
  </body>
</html>
```

[hora_01.html](#)

Un cop escrit o editat el contingut del fitxer `hora_01.html` que acabem de crear.

Guardem el document **CTRL+s**

I passem a comprovar quin és el contingut de la nostra carpeta.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ ls -la
```

```
total 6
drwxr-xr-x 1 root root  0 Sep 19 18:12 ./
drwxr-xr-x 1 root root  0 Sep 19 16:23 ../
drwxr-xr-x 1 root root  0 Sep 19 18:05 .git/
-rw-r--r-- 1 root root 18 Sep 19 16:58 README.md
-rw-r--r-- 1 root root 300 Sep 19 18:12 hora_01.html
```

A continuació, comprovarem quin és l'estat en què es troba el nostre **repositori**, amb l'ordre **git status**.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ git status
On branch main

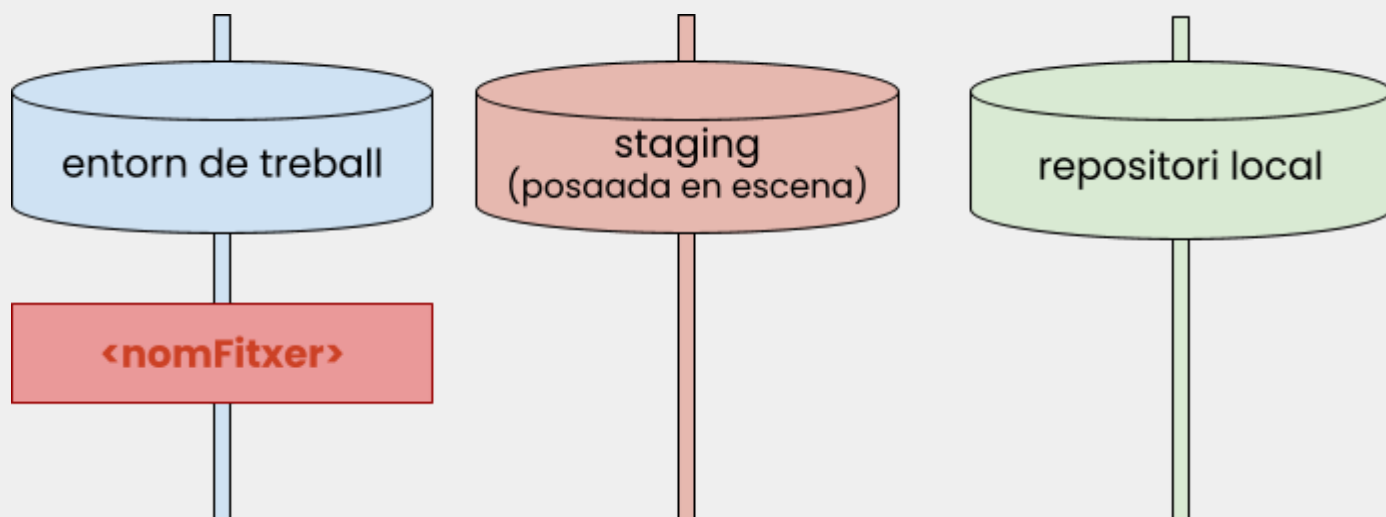
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md
    hora_01.html

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

El resultat de l'execució ens indica que l'estat del repositori és el següent:

- 1) Tenim un fitxer **hora_01.html** que es troba en un estat **Untracked files** (fitxers sense rastrejar).



Els fitxers que es troben en un estat **Untracked files** (fitxers sense rastrejar), són aquells fitxers que **git** no té a la seva llista de **fitxers a controlar**.

Per tant, perquè aquest fitxer aparegui a la llista de **git** dels **fitxers a controlar**, cal que l'afegim. Per fer-ho, cal fer servir l'ordre:

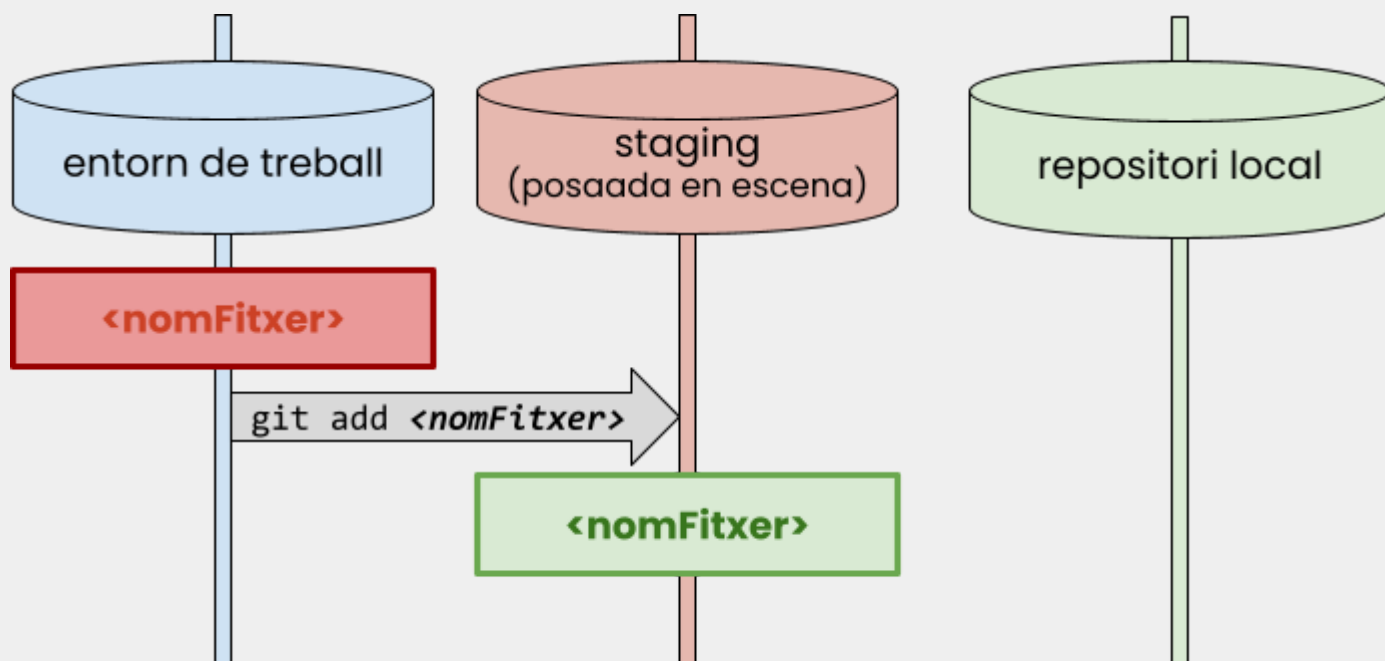
Sintaxi de l'ordre `git add <nomFitxer>`

```
git add [<nomFitxer>...]
```

Afegim el fitxer `hora_01.html` la llista de **fitxers a controlar**.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ git add hora_01.html
```

Tornem a comprovar en quin estat s troba el repositori.



```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
```

```
$ git status
```

```
On branch main
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   hora_01.html
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
README.md
```

En aquests moments, el fitxer `hora_01.html` ja es troba dins de la secció

Changes to be committed (**Canvis a cometre o realitzar**), és a dir, que es troba a la llista de **fitxers a controlar** que té **git**.

El següent pas serà fer un **commit** per crear un punt de restauració (una foto, una instantània, un **snapshot**) al qual podrem tornar, sempre que ho necessitem. Per fer-ho, cal fer servir l'ordre **git commit**.

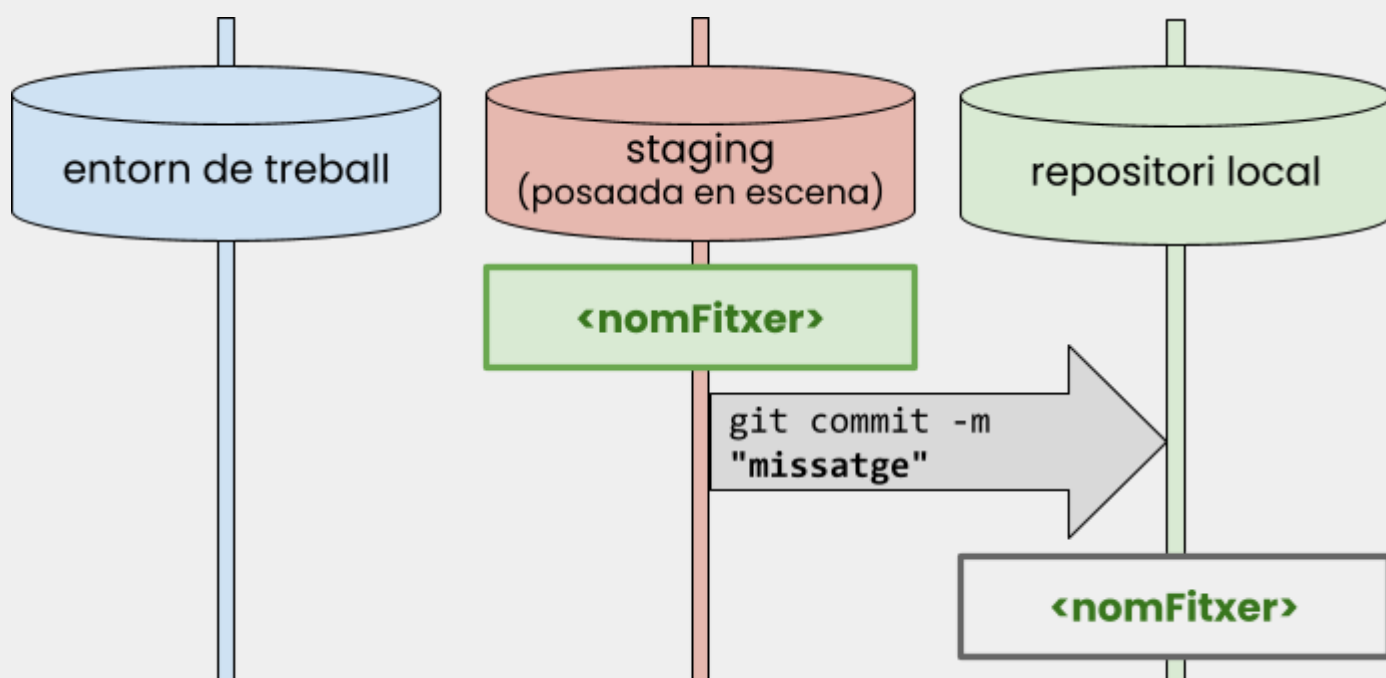
Sintaxi de l'ordre `git commit`

```
git commit [-m <missatge>] [<nomFitxer>...]
```

És **molt important** que cada **commit** tingui un missatge amb la descripció dels canvis realitzats des del darrer **commit**. Aquesta importància rau en el fet que quan vulguem escollir entre tots els **commit** que tenim, l'única informació que tindrem de cada **commit** serà aquest missatge. Per tant, cal que sigui prou descriptiu per diferenciar-lo entre la resta de **commits** que tindrem.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ git commit -m "Creem i afegim el fitxer hora_01.html"
[main (root-commit) 223f7af] Creem i afegim el fitxer hora_01.html
1 file changed, 12 insertions(+)
create mode 100644 hora_01.html
```

L'ordre **git commit**, té molts paràmetres, però hi ha un de particular.



git commit -a

El paràmetre **-a** (**--all**) especifica a l'ordre **git commit** que afegeixi automàticament tots els fitxers que s'han modificat i suprimit, llevat dels fitxers **nous**, que com que **git** no n'està informat no es veuen afectats.

Abans de provar-ho, recordem l'estat del nostre repositori.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ git status
On branch main

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

Ja que si tots aquells fitxers que es trobin en un estat **Untracked files** (**fitxers sense rastrejar**), amb el paràmetre **-a** (**--all**) no entraran al **commit**.

Sintaxi de l'ordre [git log](#)

git log [--oneline]

Per veure els **commits** que existeixen al nostre repositori, cal fer servir la comanda **git log**.

Si executem aquesta l'ordre **git log**.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
```

```
$ git log
commit 223f7af8c6d9732b6692dd895e827f140931c8f6 (HEAD -> main)
Author: joanpardogine <joanpardo@ginebro.cat>
Date: Wed Apr 19 15:00:00 2017 +0200
```

Creem i afegim el fitxer hora_01.html

En aquests moments ja tenim el nostre primer **commit**, és a dir, ja tenim un punt de restauració, al que podrem tornar, sempre que vulguem.

És interessant comentar que l'ordre **git log**, té diferents paràmetres, però hi ha un que és prou útil.

```
root@Win2354 MINGW64 /d/SXM2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ git log --oneline
223f7af (HEAD -> main) Creem i afegim el fitxer hora_01.html
```

D'aquesta manera quan hi hagi molts **commits** serà més fàcil trobar el que ens interessa.

Per veure els fitxers que hi ha en un **commit**, cal fer servir la comanda **git log**.

Sintaxi de l'ordre [git ls-tree](#)

```
git ls-tree --name-only -r <identificadorDelCommit>
```

Per exemple, per veure els fitxers que hi ha en un **commit**, que té l'identificador

223f7af cal executar el següent:

```
root@Win2354 MINGW64 /d/SXM2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ git ls-tree --name-only -r 223f7af
hora_01.html
```

Per últim, abans de pujar el nostre repositori local al nostre repositori remot allotjat a **github.com**, crearem un segon **commit**.

Recordem que els fitxers que fem servir els podeu trobar a:

- [hora_01.html](#) i [hora_01.js](#)
- [hora_02.html](#) i [hora_02.js](#)
- [hora_03.html](#) i [hora_03.js](#)

Per crear un segon **commit**, procedirem a crear un fitxer anomenat **hora_01.js**, amb el contingut del fitxer **hora_01.js**, o simplement copiarem el fitxer a la carpeta del nostre repositori.

Un cop creat o copiat el fitxer **hora_01.js**, mirem quin és el contingut de la carpeta del nostre repositori.

```
root@Win2354 MINGW64 /d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U
(main)
$ ls -la
total 6
drwxr-xr-x 1 root root  0 Sep 19 18:12 ./
drwxr-xr-x 1 root root  0 Sep 19 16:23 ../
drwxr-xr-x 1 root root  0 Sep 19 18:05 .git/
-rw-r--r-- 1 root root 18 Sep 19 16:58 README.md
-rw-r--r-- 1 root root 300 Sep 19 18:12 hora_01.html
```

A continuació, comprovarem quin és l'estat en què es troba el nostre **repositori**, amb l'ordre **git status**.

```
root@Win2354 MINGW64
/d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U (main)
```

```
$ git status
On branch main
```

Untracked files:

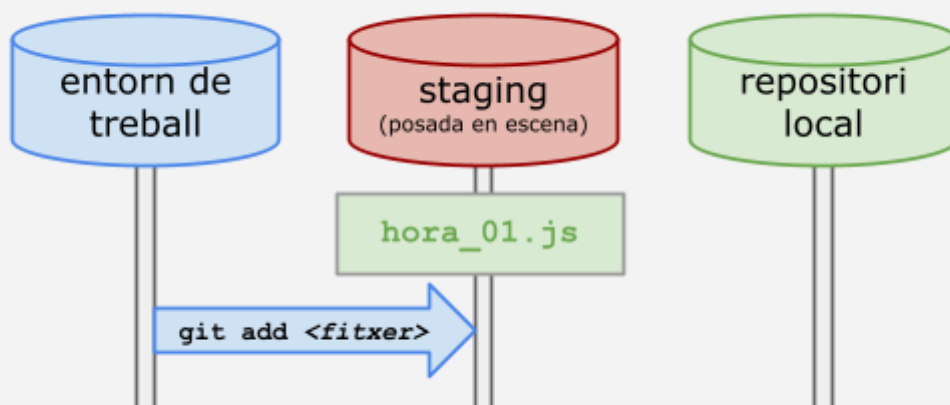
(use "git add <file>..." to include in what will be committed)

hora_01.js

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

El resultat de l'execució ens indica que l'estat del repositori és el següent:

- 1) Tenim un fitxer **hora_01.js** que es troba en un estat **Untracked files** (fitxers sense rastrejar) i
- 2) de la resta no ens diu res, ja que l'únic fitxer que hi ha, a part del fitxer **hora_01.js** és el fitxer **hora_01.html**, que **git** el té controlat, i com que el seu contingut no ha estat modificat des del darrer **commit** en el que va ser modificat, **git** ni tan sols els el mostra.



En aquest pas afegirem el fitxer **hora_01.js** a la llista de **fitxers a controlar**:

```
root@Win2354 MINGW64
/d/SXM2-2022-2023/MP07/<CognomAlumne>-A015U (main)
```

```
$ git add hora_01.js
```

```
root@Win2354 MINGW64
```

```
/d/SMX2-2022-2023/MP07/<CognomAlumne>-A015U (main)
```

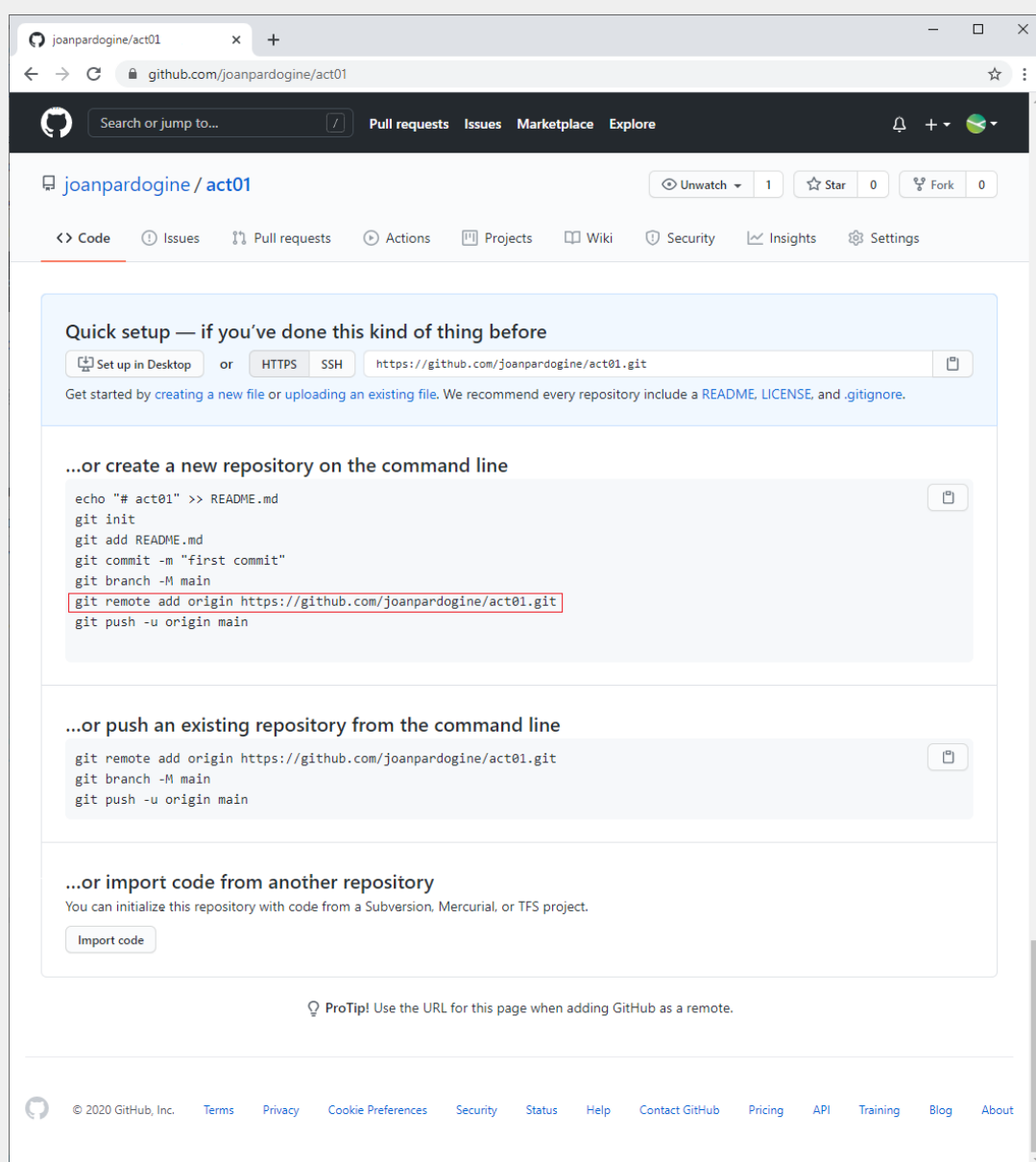
```
$ git status
```

```
On branch main
```

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

```
new file:   hora_01.js
```



Per veure els **commits** que existeixen al nostre repositori, cal fer servir la comanda **git log**.

Per poder-lo pujar al **github.com** cal crear tenir creat el repositori.

En el nostre cas, és el que ja hem creat amb el nom **<CognomAlumne>-A015U**

Un cop creat, el repositori anomenat **<CognomAlumne>-A015U**, procedirem a pujar el nostre repositori.

La comanda per poder vincular el nostre repositori local amb el nostre repositori de github.com és:

```
git remote add [-t <branch>] [-m <main>] [-f] [--[no-]tags]
[--mirror=(fetch|push)] <name> <url>
```

```
root@Win2354 MINGW64
/d/SXM2-2022-2023/MP07/<CognomAlumne>-A015U (main)
$ git remote add origin
https://github.com/<nomUsuariGithub>/<CognomAlumne>-A015U.git
```

La comanda per "pujar" el nostre repositori local cap al nostre repositori de github.com és:

```
root@Win2354 MINGW64
/d/SXM2-2022-2023/MP07/<CognomAlumne>-A015U (main)
$ git push -u origin main
```

Lliurament

Cal que creeu un fitxer amb el nom **README.md** al vostre repositori amb tots els passos que heu fet per assolir el que demana l'activitat. Cal que de cada pas escriviu **la comanda**, així com el **text resultant** de l'execució.