

## Activitat 2: GitHub

### Introducció

En aquesta activitat aprendrem a utilitzar les facilitats de gestió de projectes de GitHub.

Per a fer aquesta activitat es recomana l'ús de Ubuntu LTS (p.e. 22.04).

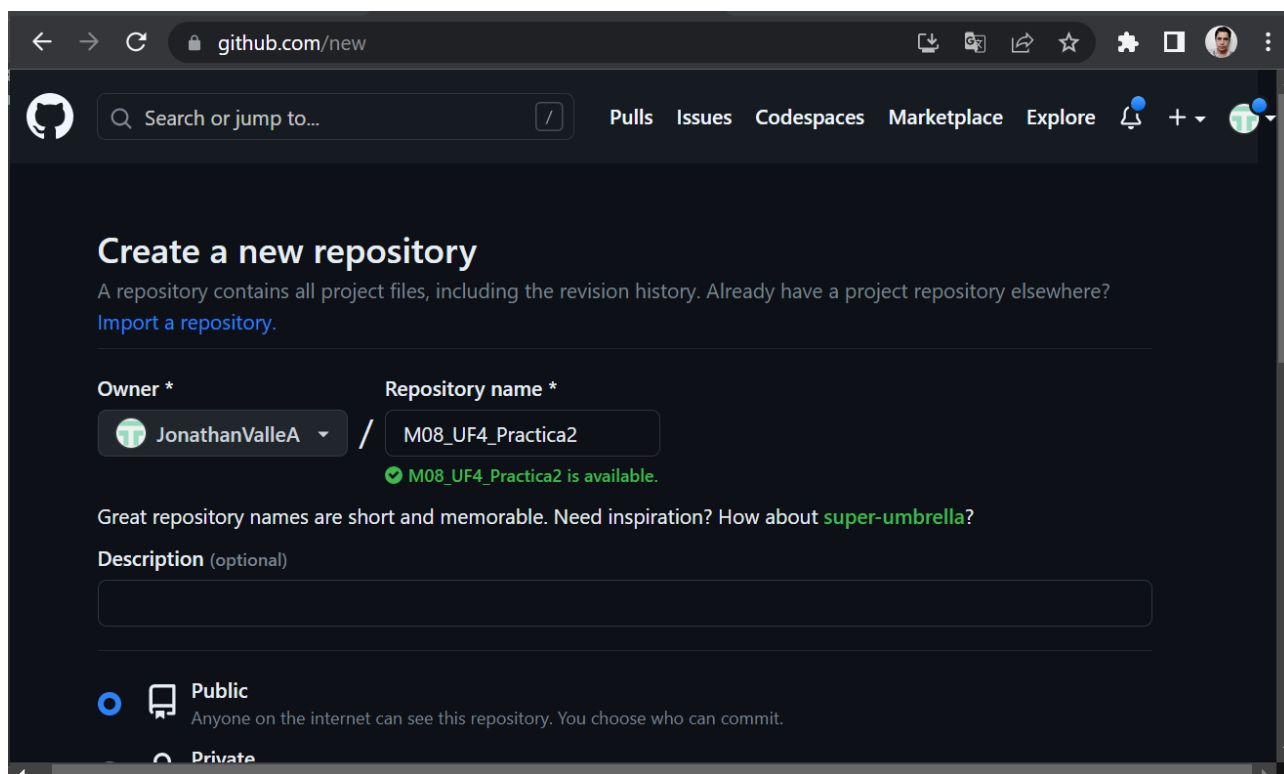
Hauràs de fer captures de cada pas i retallar-les per tal que es vegin bé en una còpia d'aquest document que hauràs de lliurar de l'activitat.

### Explorant les possibilitats de GitHub

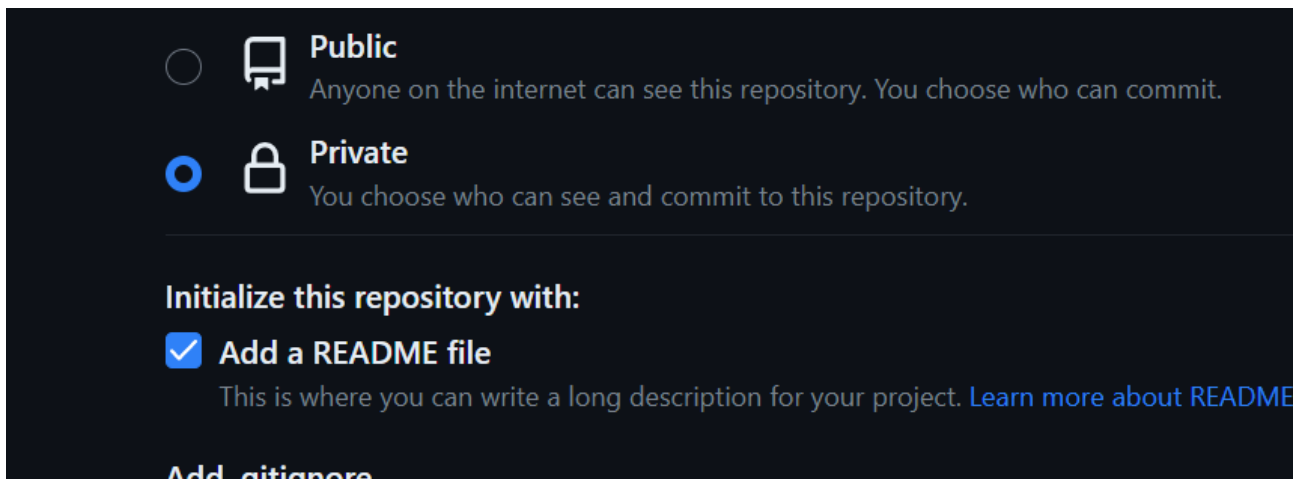
L'objectiu és simular el treball habitual de gestió d'un repositori de GitHub quan es treballa en equip.

1. Crea un repositori a GitHub amb el teu compte

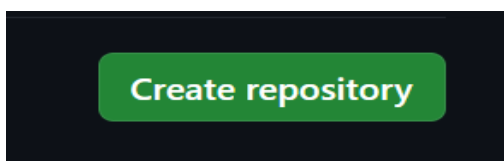
Dentro de mi cuenta de GitHub, creó un nuevo repositorio con el nombre M08\_UF4\_Practica2.



También se puede definir que tu repositorio sea Público o Privada, en mi caso lo hago Privado ya que es un repositorio de prueba y no quiero que nada del exterior lo pueda ver (únicamente pueden acceder a este repositorio si le doy acceso) y generamos un archivo README para que de esta forma se cree un repositorio y se pueda aplicar el git clone, sino el repositorio estará vacío y habrá que hacer un git init.

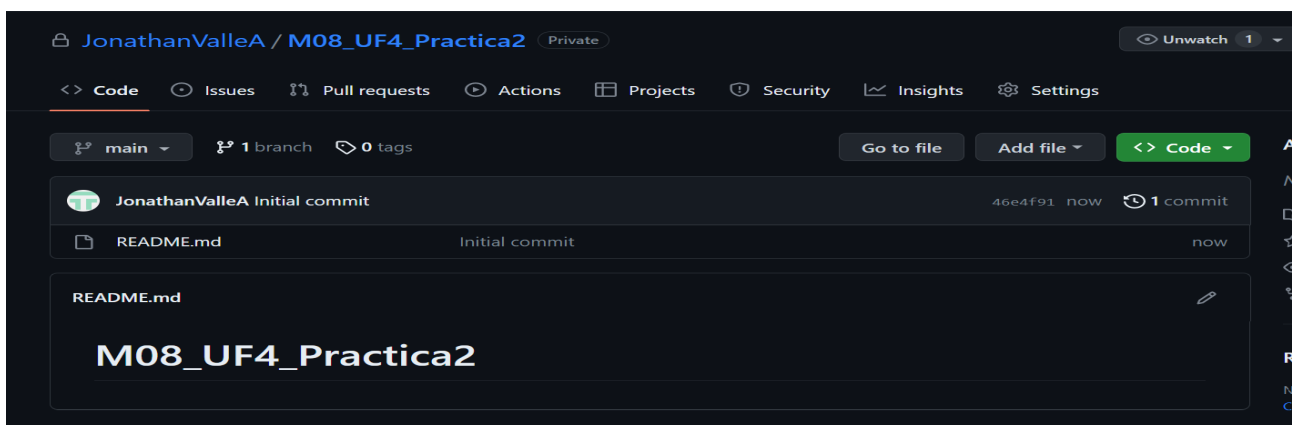


Se crea el repositorio para ya poder comenzar a trabajarlo.

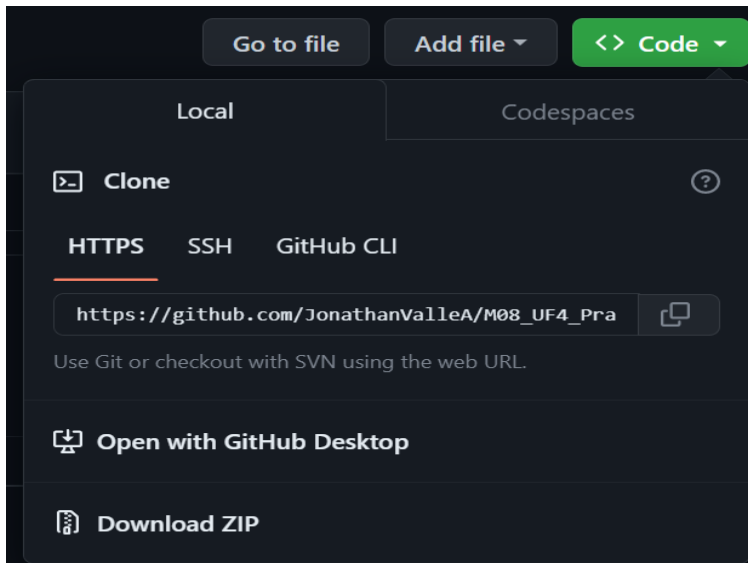


## 2. Inicialitza una carpeta local amb el repositori remot (git clone ...)

Podemos visualizar que ya está creado correctamente el repositorio con un archivo README.



En el Code, podemos ver la url de mi repositorio, únicamente se copia, también se puede con la url de la página.



En el CMD, se hace un git clone “y la url del repositorio” y si todo ha ido bien se creara tu carpeta.

```
C:\Users\jonat>git clone https://github.com/JonathanValleA/M08_UF4_Practica2.git
Cloning into 'M08_UF4_Practica2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Se creará una carpeta con el mismo nombre que el repositorio, si accedemos a esa carpeta estara el archivo README.md creado.

```
C:\Users\jonat>cd M08_UF4_Practica2
C:\Users\jonat\M08_UF4_Practica2>dir
El volumen de la unidad C es OS
El número de serie del volumen es: AA21-AB36

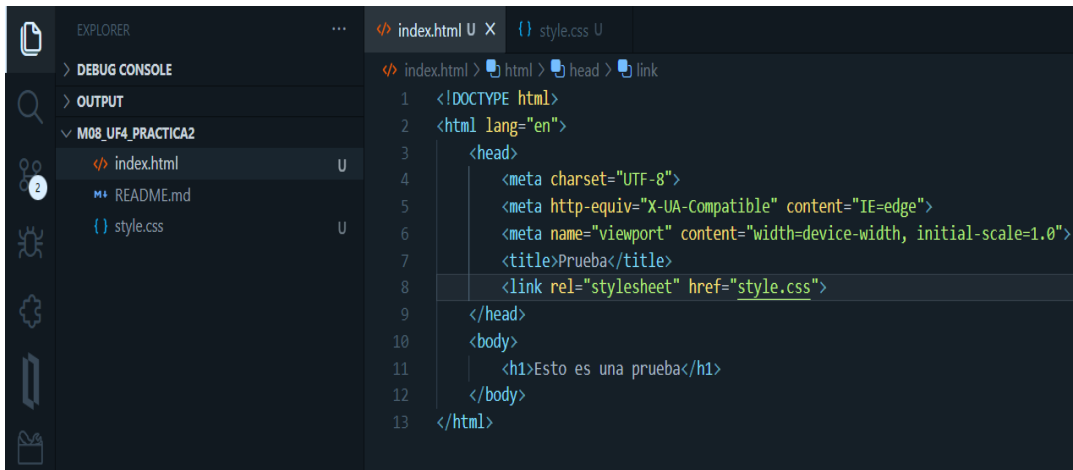
Directorio de C:\Users\jonat\M08_UF4_Practica2

17/05/2023  17:53    <DIR>          .
17/05/2023  17:53    <DIR>          ..
17/05/2023  17:53                19 README.md
                1 archivos                19 bytes
                2 dirs 26.636.378.112 bytes libres

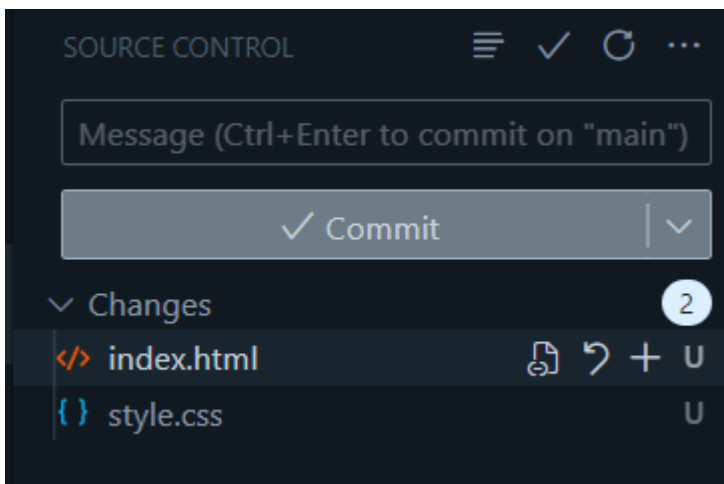
C:\Users\jonat\M08_UF4_Practica2>
```

## 3. Afegeix uns quants arxius al repositori local i fes commit

En el visual, agregue 2 archivos index.html y style.css



A la izquierda del visual, en Source Control se podrán ver todos los cambios que se han realizado, en este caso pues son la creación de esos 2 archivos y el contenido dentro.



En el CMD, también es otra forma de ver los archivos creados.

```
C:\Users\jonat\M08_UF4_Practica2>dir
El volumen de la unidad C es OS
El número de serie del volumen es: AA21-AB36

Directorio de C:\Users\jonat\M08_UF4_Practica2

17/05/2023  17:56    <DIR>          .
17/05/2023  17:53    <DIR>          ..
17/05/2023  17:56                385 index.html
17/05/2023  17:53                19 README.md
17/05/2023  17:56                39 style.css
               3 archivos                443 bytes
               2 dirs 26.626.420.736 bytes libres

C:\Users\jonat\M08_UF4_Practica2>
```

Ahora se sube los cambios al GitHub, Para ello se ejecuta el comando git add y el git commit -m para poner un mensaje al commit.

```
C:\Users\jonat\M08_UF4_Practica2>git add .
C:\Users\jonat\M08_UF4_Practica2>git commit -m "Archivos index.html y style.css"
[main 0247676] Archivos index.html y style.css
2 files changed, 16 insertions(+)
create mode 100644 index.html
create mode 100644 style.css
```

#### 4. Puja'ls al repositori GitHub

Se hace un git push para subir los archivos al repositorio de GitHub. Como se puede ver se ha subido a la url que te indica y a la rama main.

```
C:\Users\jonat\M08_UF4_Practica2>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 603 bytes | 301.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JonathanValleA/M08_UF4_Practica2.git
46e4f91..0247676  main -> main

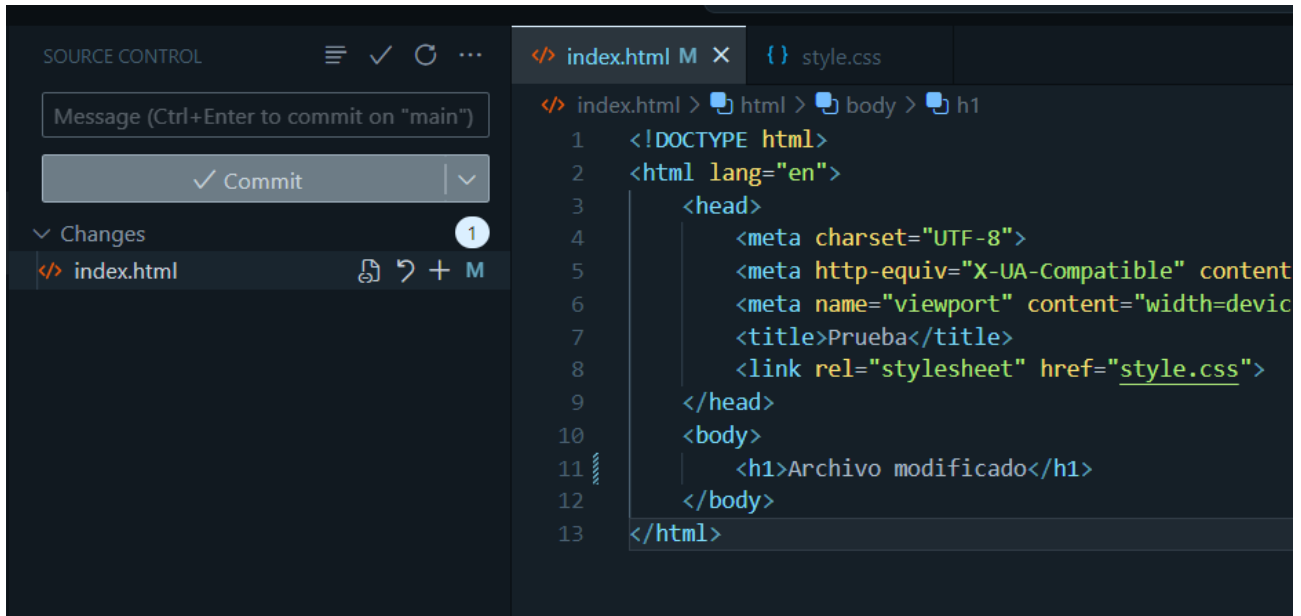
C:\Users\jonat\M08_UF4_Practica2>
```

5. Què és una petició d'extracció o Pull Request?

Es un commit o merge que se hace hacia un repositorio remoto, donde el dueño de ese repositorio debe aprobarlo, es decir, subes tus archivos al GitHub a tu rama que esta apartada del main y con el pull request es para aprobar esos cambios que has hecho y unificarlo hacia la main o master (rama principal), básicamente se utiliza entre desarrolladores para trabajar en grupo

6. Crea una pull request en el teu repositori modificant un arxiu que contingui codi (HTML, JavaScript, Java, PHP, Python, etc...)

Ahora como puedes ver, he modificado el archivo index.html y en Source Control me aparece que ha habido un cambio y que puedo hacer un commit.



Creo una nueva rama donde subire todo de nuevo con la modificación que he hecho.

```
C:\Users\jonat\M08_UF4_Practica2>git checkout -b branch_jonathan
Switched to a new branch 'branch_jonathan'

C:\Users\jonat\M08_UF4_Practica2>git branch
* branch_jonathan
  main

C:\Users\jonat\M08_UF4_Practica2>
```

Procedo a subir los cambios a mi nueva rama para luego realizar un pull request hacia la main.


```
C:\Users\jonat\M08_UF4_Practica2>git push
fatal: The current branch branch_jonathan has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin branch_jonathan

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

C:\Users\jonat\M08_UF4_Practica2>git push --set-upstream origin branch_jonathan
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch_jonathan' on GitHub by visiting:
remote:   https://github.com/JonathanValleA/M08_UF4_Practica2/pull/new/branch_jonathan
remote:
To https://github.com/JonathanValleA/M08_UF4_Practica2.git
 * [new branch]      branch_jonathan -> branch_jonathan
branch 'branch_jonathan' set up to track 'origin/branch_jonathan'.

C:\Users\jonat\M08_UF4_Practica2>
```

 branch\_jonathan had recent pushes less than a minute ago [Compare & pull request](#)

branch\_jonathan ▾

2 branches



0 tags




Go to file

Add file ▾

<> Code ▾

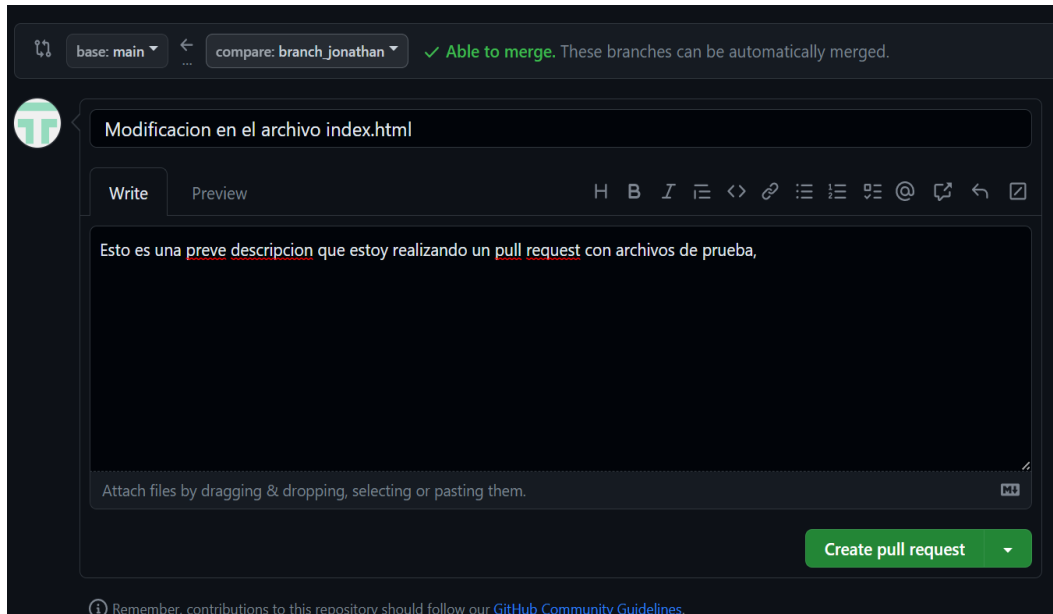
This branch is 1 commit ahead of main. [Contribute ▾](#)

 JonathanValleA Modificacion en el archivo index.html dd6dd46 now  3 commits

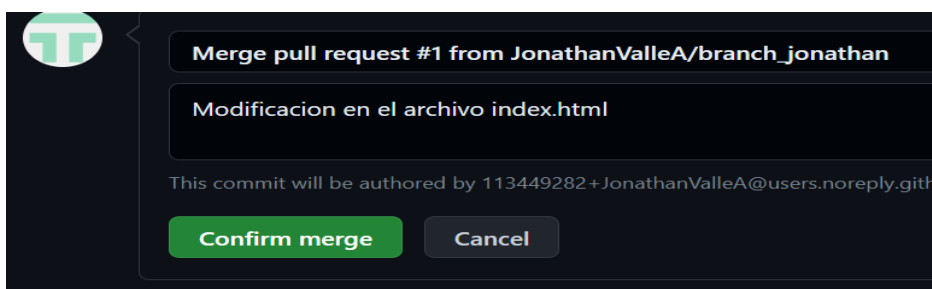
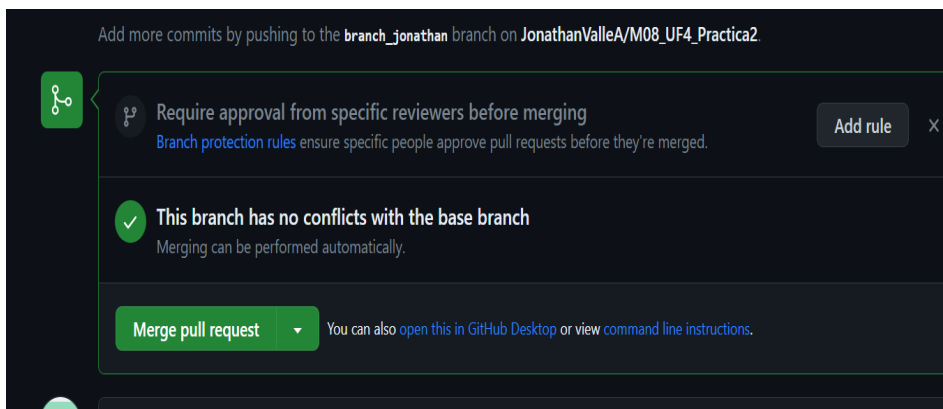
 README.md	Initial commit	40 minutes ago
 index.html	Modificacion en el archivo index.html	now
 style.css	Archivos index.html y style.css	33 minutes ago

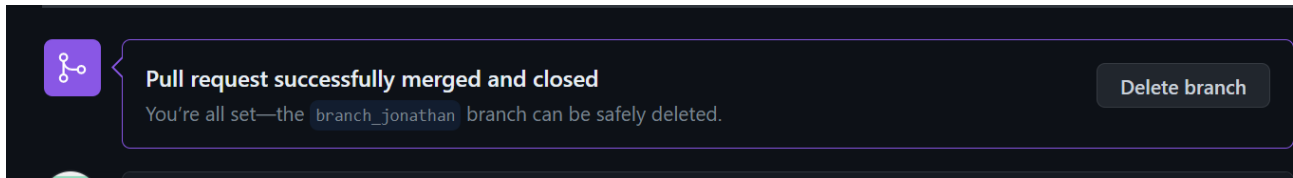
7. Fes comentaris a la pull request introduint alguna suggerència de millora de la pull request.









## 8. Accepta la pull request i fes el merge





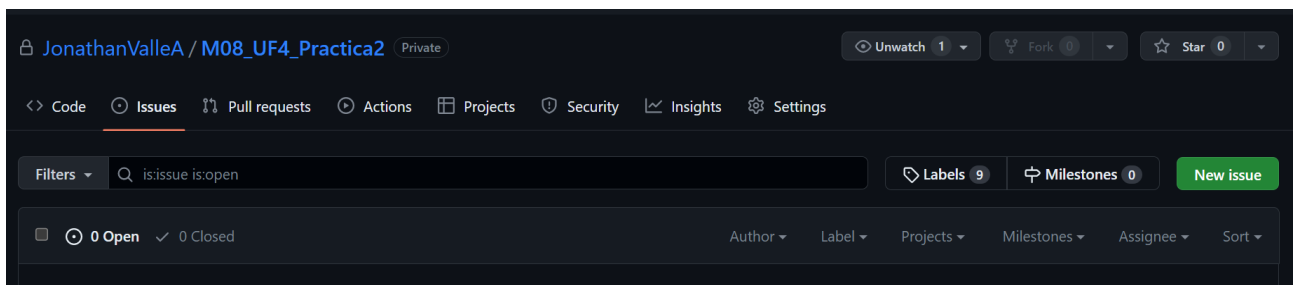
Como se puede ver a la hora de hacer el pull request, en la main se podrán ver los cambios reflejados.

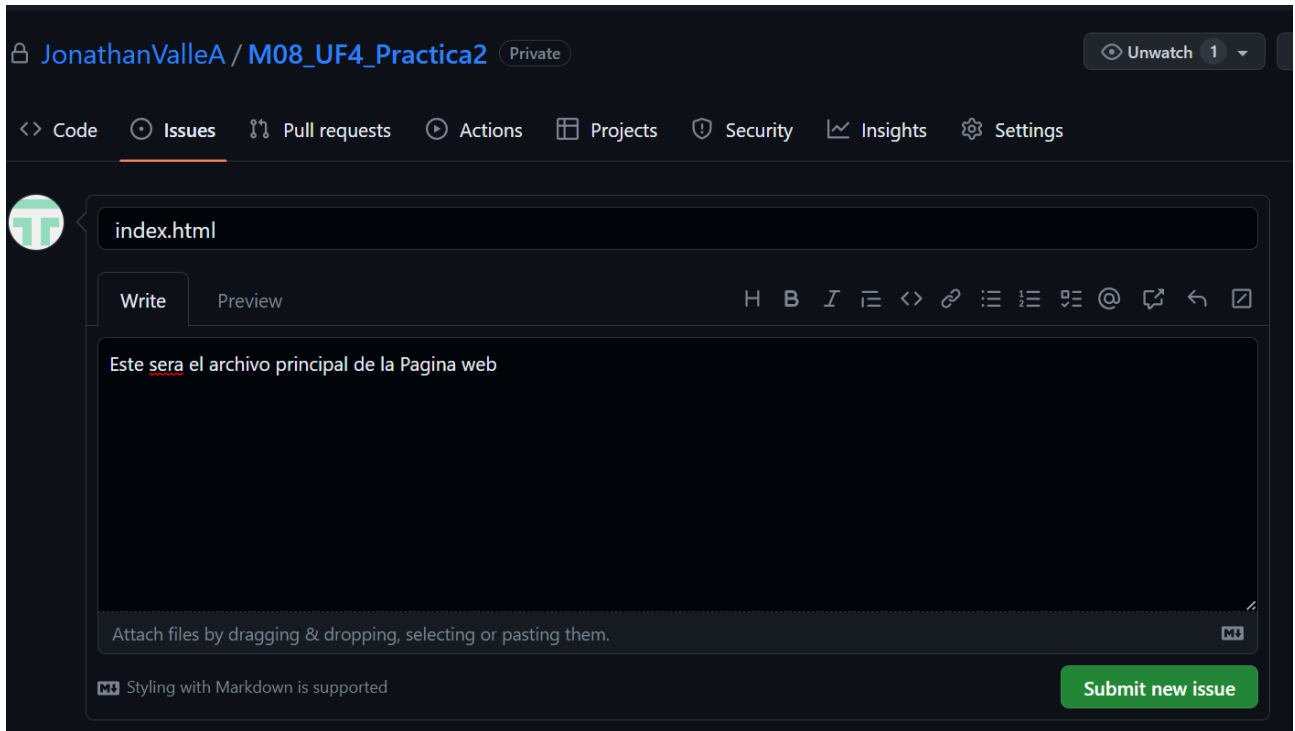
	JonathanValleA Merge pull request #1 from JonathanValleA/branch_jonathan	dc69529 now	🕒 4 commits
	README.md	Initial commit	47 minutes ago
	index.html	Modificacion en el archivo index.html	7 minutes ago
	style.css	Archivos index.html y style.css	41 minutes ago

### 9. Què és un assumpte o issue en un projecte GitHub?

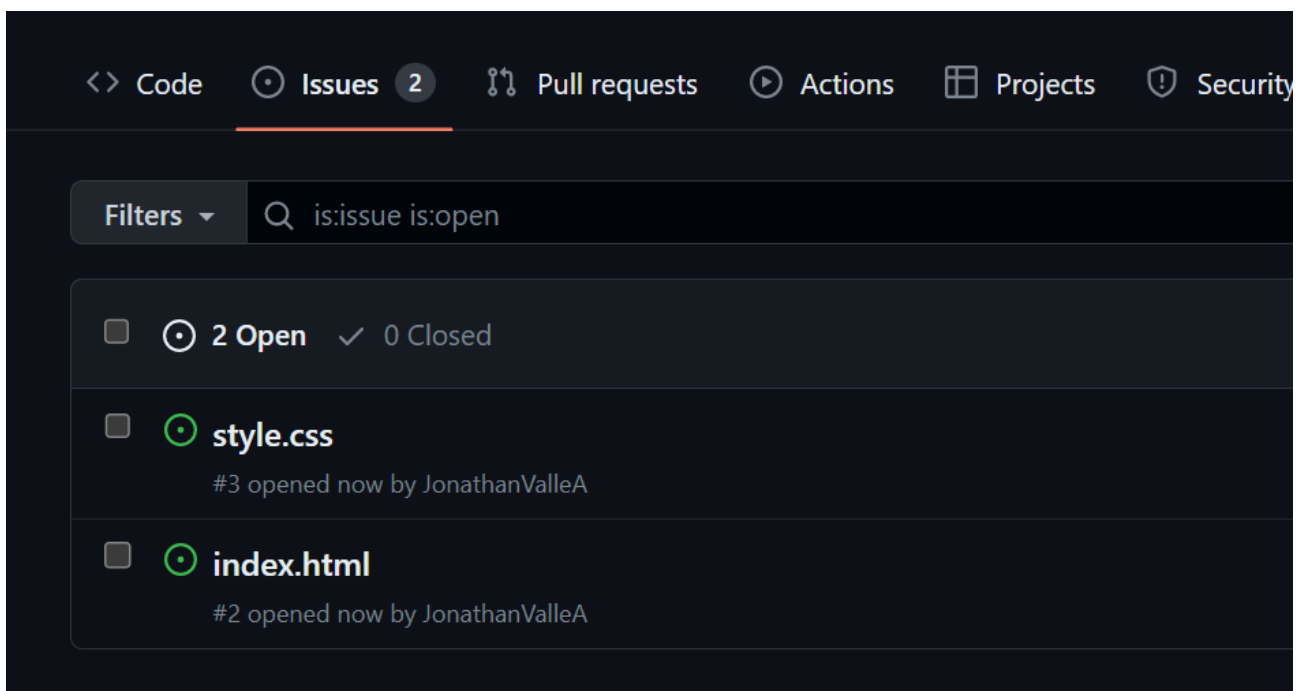
Es una nota de repositorio que trata de llamar la atención sobre un problema. Puede ser un error a corregir, una petición para añadir, una nueva opción o característica, una pregunta para aclarar algún tema que no está correctamente aclarado o muchas otras cosas diferentes.

### 10. Crea diversos issues i dona a cadascun una etiqueta predefinida diferent



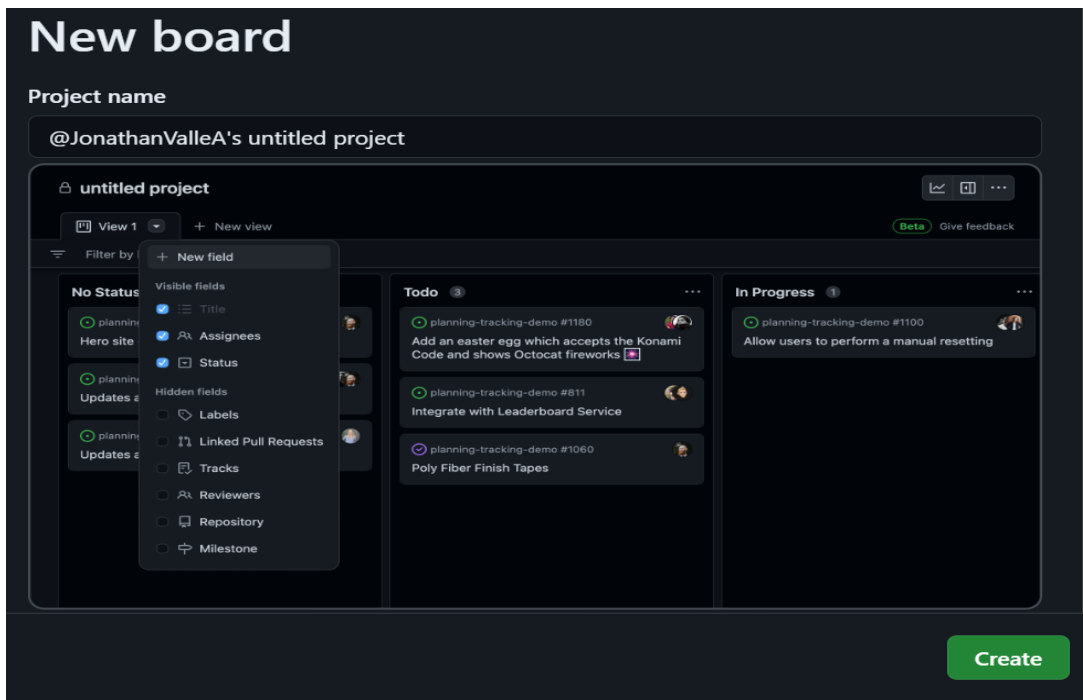
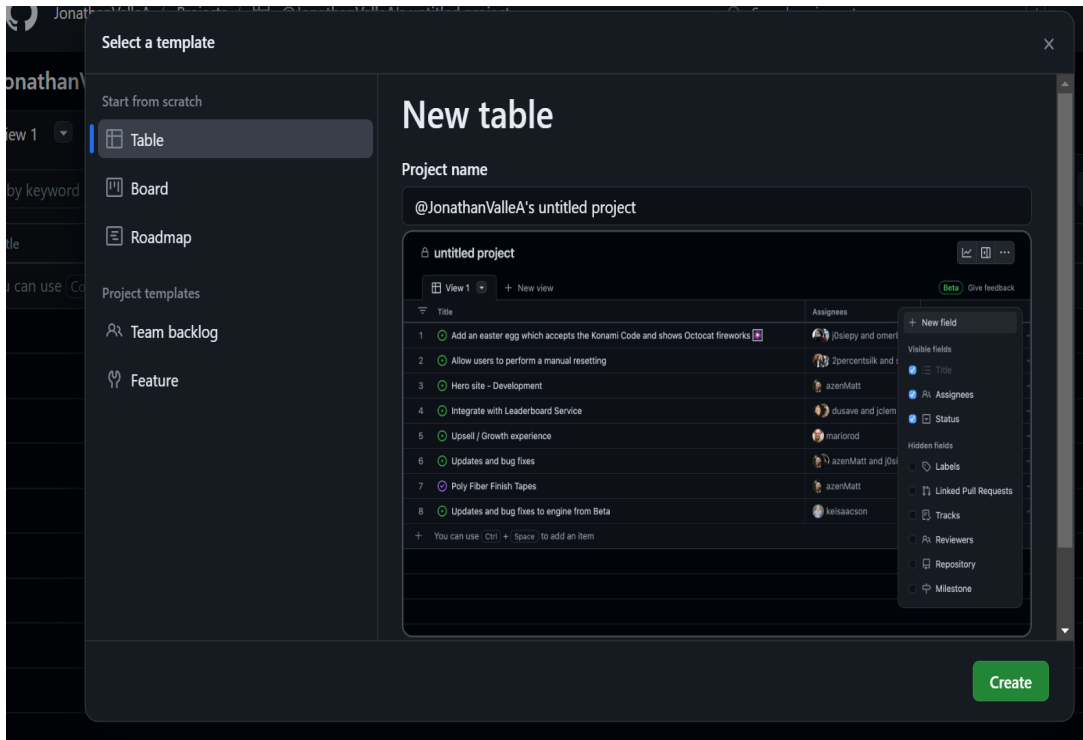


Únicamente cree 2 issues para los 2 archivos definidos.

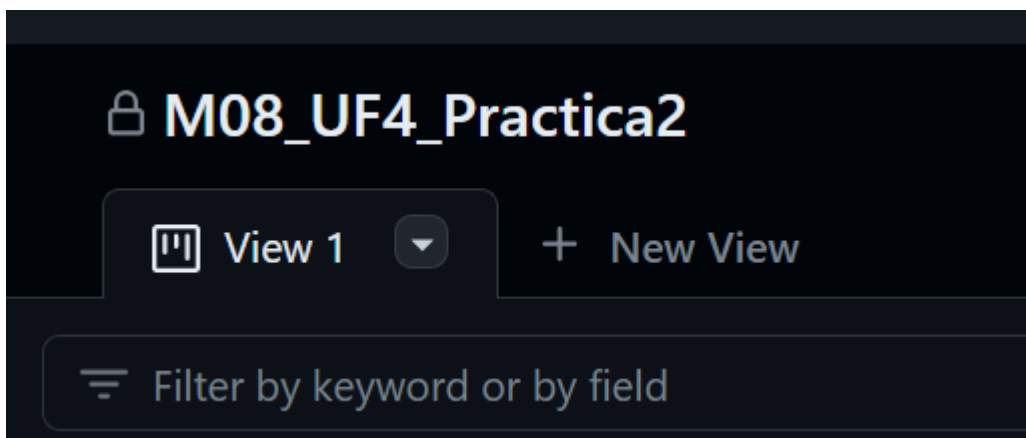
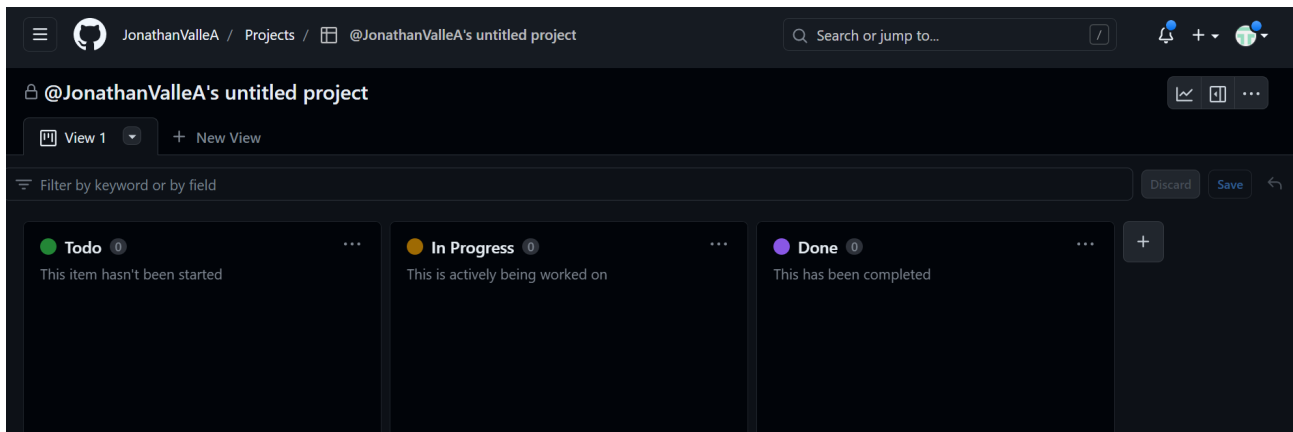


11. Revisa el concepte de projecte dins de la pestanya Projects. Crea un projecte i utilitza la plantilla de Simple Kanban

Nos dirigimos al apartado Project y al crear un nuevo proyecto se abra un menu para elegir el tipo de proyecto o Kanban que deseamos tener.



El Kanban ya creado, se puede asignar otro nombre al Kanbar.

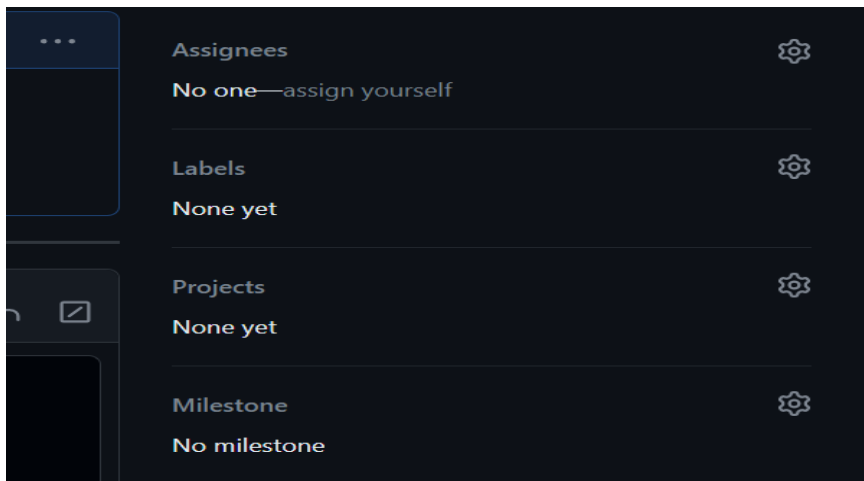


12. Investiga com es relacionen les issues amb el Kanban i explica-ho.

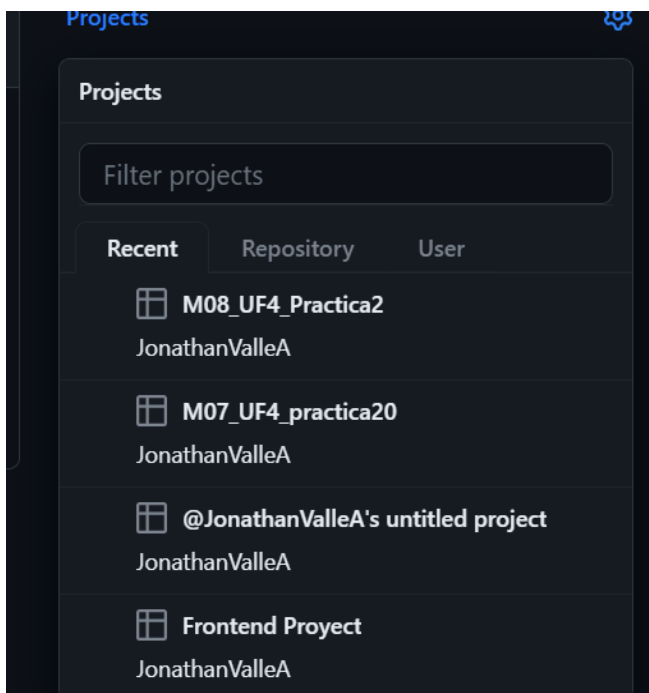
Kanban és un sistema de gestió visual que s'utilitza per organitzar i controlar el flux de treball. Les "issues" són problemes o situacions que requereixen atenció o resolució. En Kanban, les issues es representen com a targetes en un tauler i es mouen a través de les etapes per indicar el seu estat i el progrés. Això permet una visualització clara i facilita la gestió eficaç de les issues dins del sistema Kanban.

13. Ara dona d'alta una issue i assigna-la al projecte que has creat. Ves a la pestanya de projecte i mira si t'apareix en «add card». Fica-la en la columna primera (per fer → «to do»)

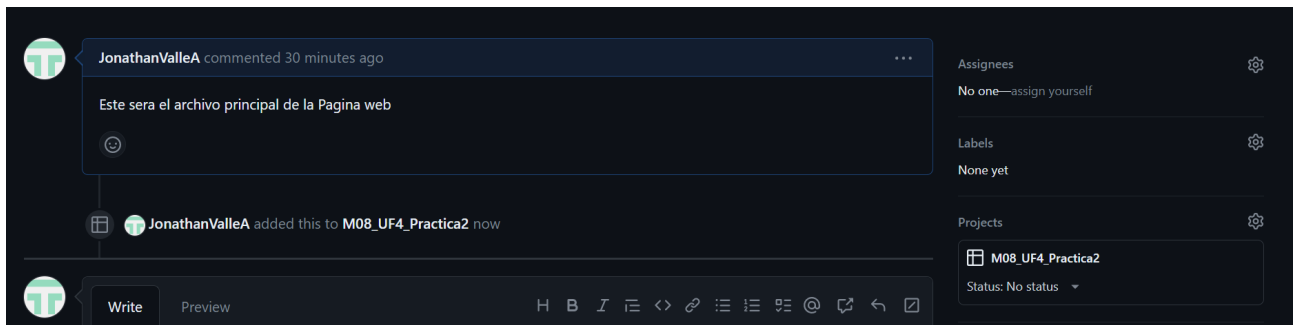
Una vez creado los issues y el Kanban, si hay que asignarlo al Kanban creado la forma de hacerlo es dirigirte a una issue y en Projects por defecto, no está asignado a ningún proyecto.



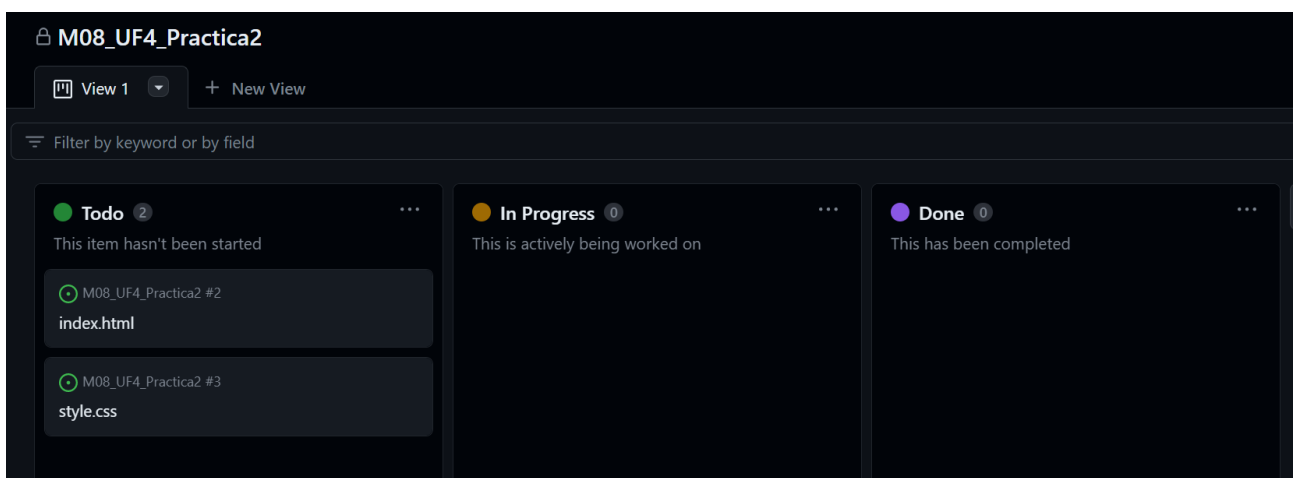
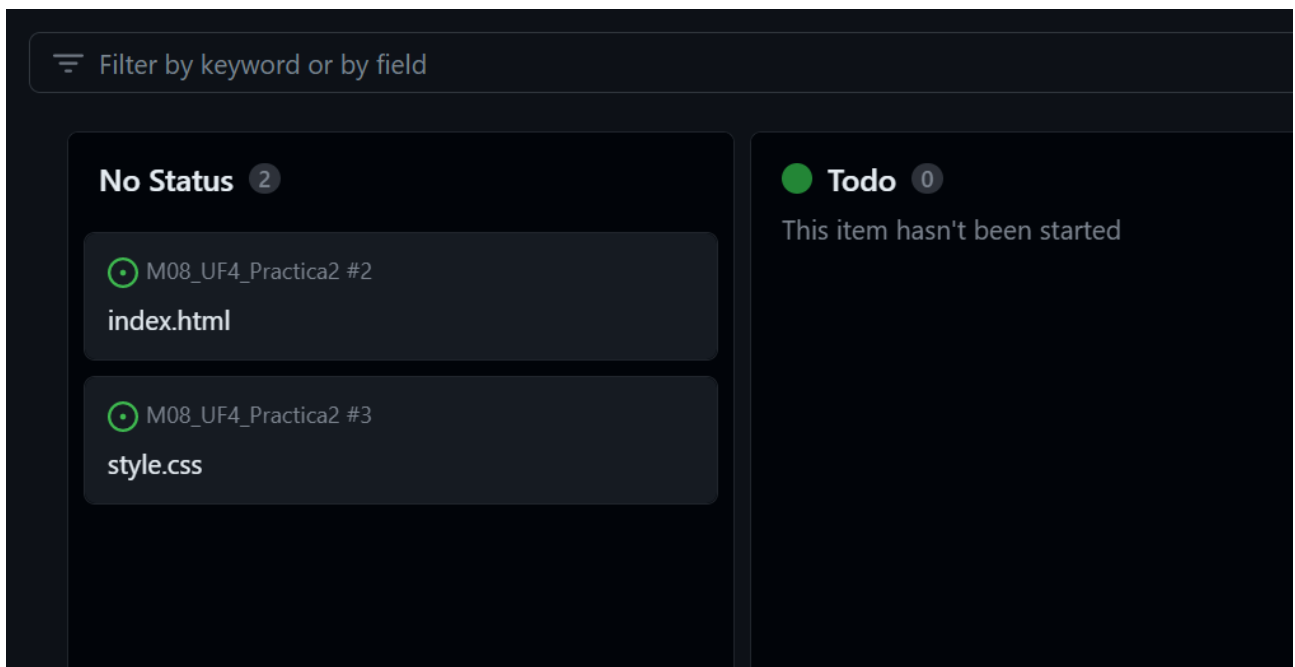
Elegimos el proyecto de nuestro repositorio, que es M08\_UF4\_Practica2.



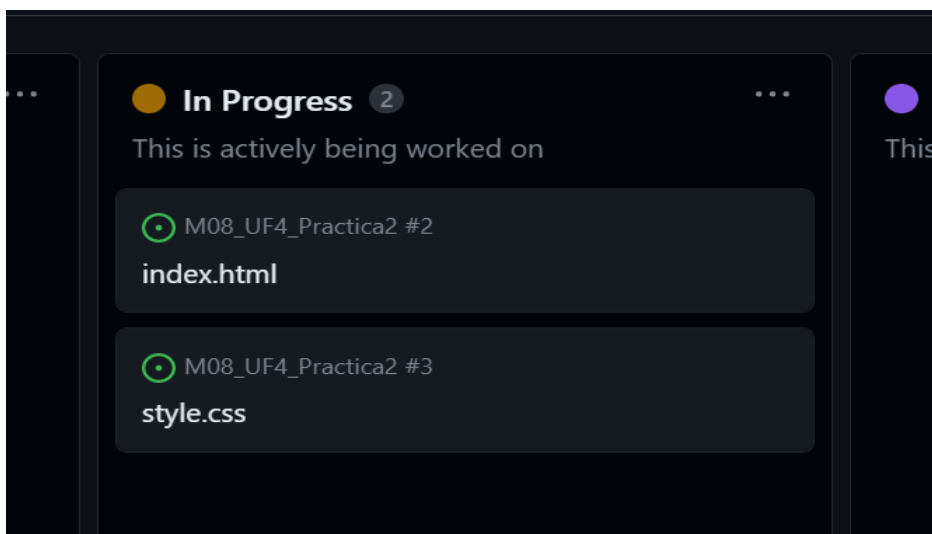
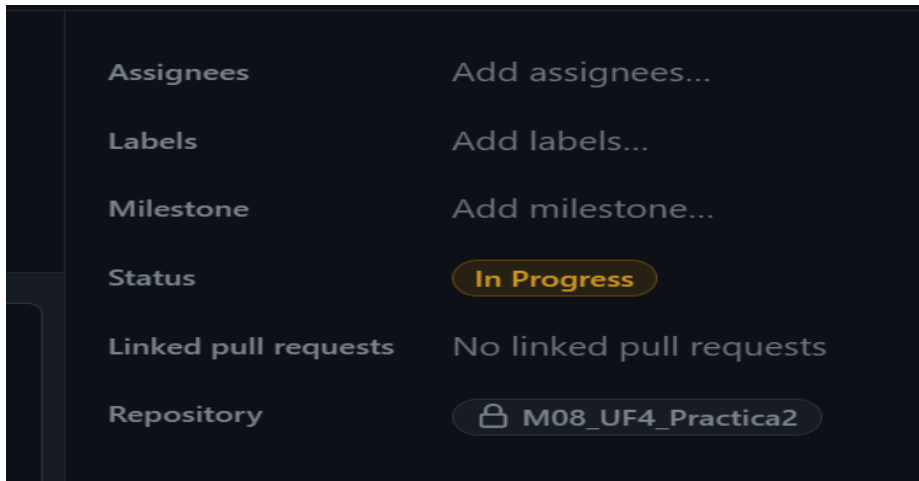
Se puede ver que ya se ha asignado está issue al Kanban y lo mismo para la otra issue.



En el Kanban ya se puede visualizar las 2 issue definidas correctamente.



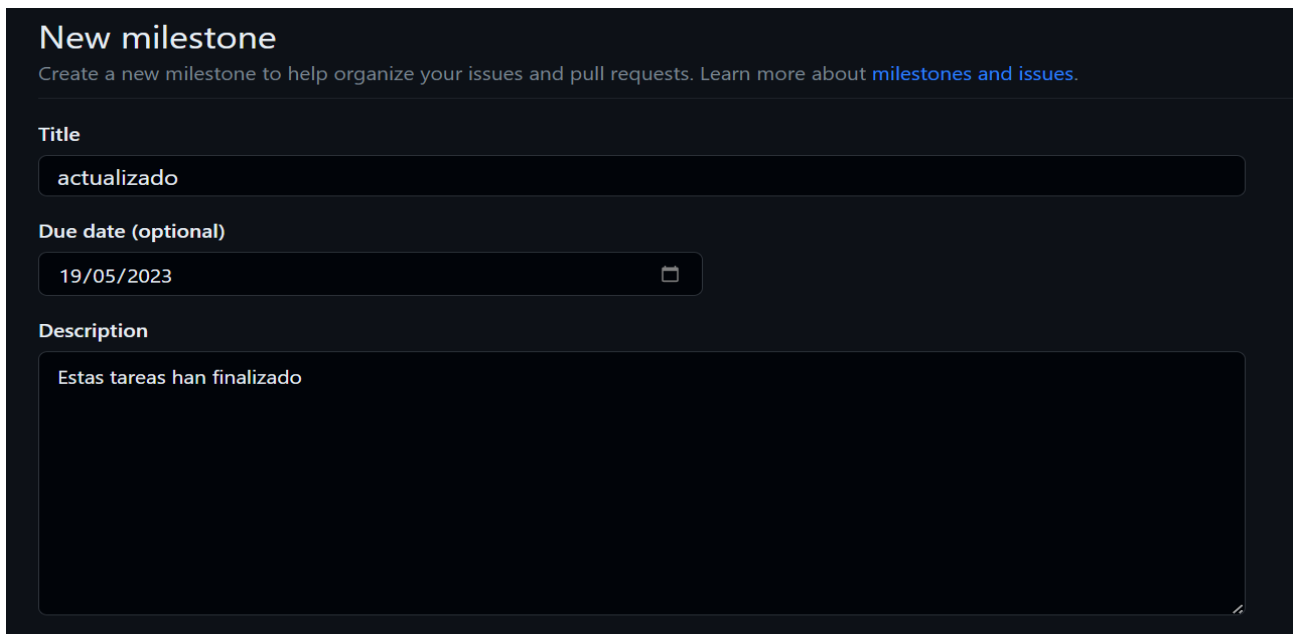
14. Ara passa la targeta de «per fer» a «in progress». Fes un canvi en el codi i en la descripció fes referència a la issue amb #número\_de\_issue



15. Canvia l'automatisme («Automation») de la columna «done» del Kanban del projecte i fes que les issues tancades i els merged pull requests vagin allí automàticament.



Primero creamos un milestone para poder agrupar o organizar tareas o issues relacionadas con el proyecto.



The screenshot shows the 'New milestone' form in GitHub. The title is 'actualizado'. The due date is '19/05/2023'. The description is 'Estas tareas han finalizado'.

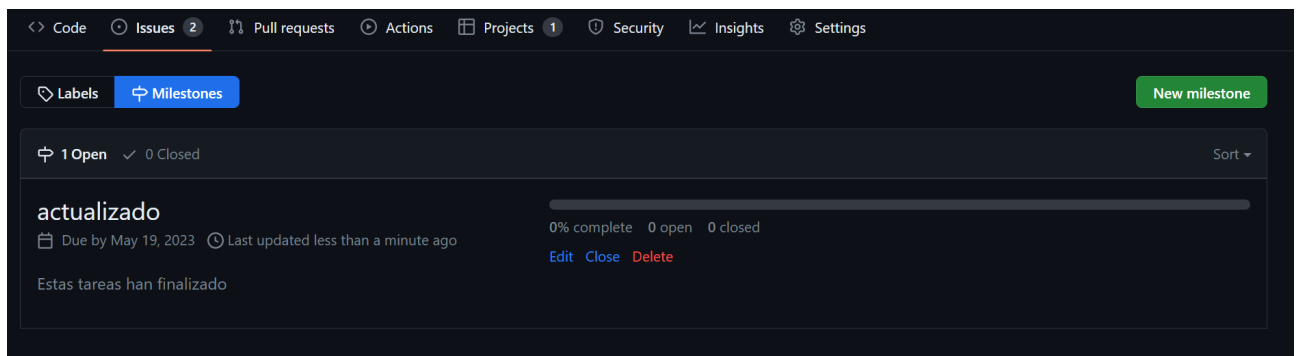
**New milestone**  
Create a new milestone to help organize your issues and pull requests. Learn more about [milestones](#) and [issues](#).

**Title**  
actualizado

**Due date (optional)**  
19/05/2023

**Description**  
Estas tareas han finalizado

Podemos ver que está todo creado, lo pasamos a closed.



The screenshot shows the GitHub Milestones page. The 'actualizado' milestone is listed with 1 Open issue and 0 Closed issues. The progress bar shows 0% complete, 0 open, and 0 closed. The description is 'Estas tareas han finalizado'.

Code Issues 2 Pull requests Actions Projects 1 Security Insights Settings

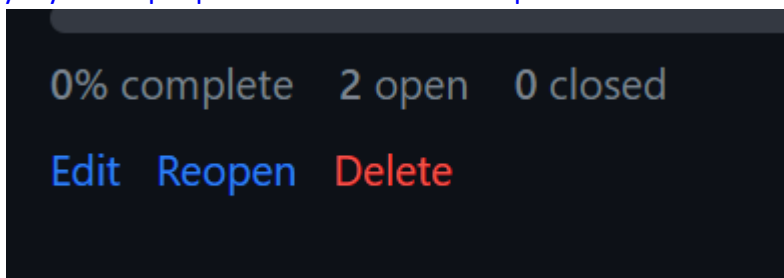
Labels Milestones New milestone

1 Open 0 Closed Sort

**actualizado**  
Due by May 19, 2023 Last updated less than a minute ago  
0% complete 0 open 0 closed  
Edit Close Delete

Estas tareas han finalizado

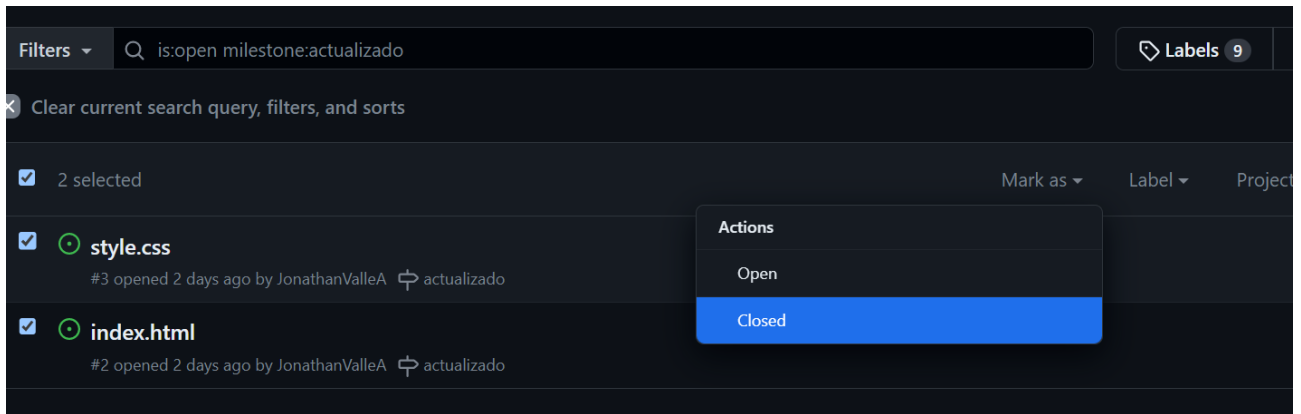
En open podemos ver todos los issues que tenemos creado que en total son 2 que es el index.html y style.css que por el momento esta en proceso



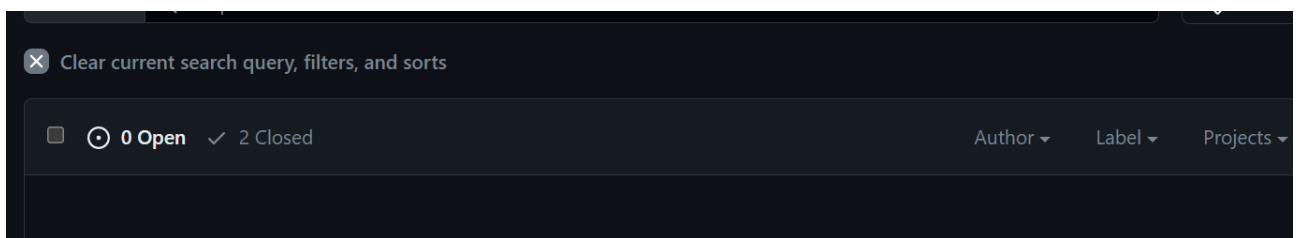
The screenshot shows the GitHub Issues page. The 'actualizado' milestone is listed with 2 Open issues and 0 Closed issues. The progress bar shows 0% complete, 2 open, and 0 closed. The description is 'Estas tareas han finalizado'.

0% complete 2 open 0 closed  
Edit Reopen Delete

Una vez visto eso, nos dirigimos a todos los issues que tenemos creado y lo seleccionamos, en Mark as habrá 2 opciones, en este caso lo queremos cerrar o finalizar estas 2 issue ya que se supone que está terminado, por lo que le damos a Closed.



Ya pasamos, podremos ver que ya hay 2 issue en Closed, osea terminados.



En el Kanban podemos ver las 2 issues que están en Done, una vez cerrado las issues se pasa directamente a Done.

