Desarrollo de Aplicaciones Web en Entorno a Cliente

Inicio

[**Documentación sobre GitHub 2**](#_fdt1gzwuge8j)

[Creación de Repository y Proyect en GitHub 2](#_69pdkjolnps)

[Tareas en Proyect 4](#_f2ddc2wkecna)

[Creación de archivos y vinculación a GitHub 5](#_bx3tmlx29sn2)

[Iniciamos Git 5](#_7rilb68fjfyl)

[Vinculación de archivos al repositorio 6](#_vs95ztgl8nen)

[**I. Proceso de ToDoList 9**](#_74ksqg6nhyvr)

[index.html 10](#_59uayd8czhkm)

[style.css 11](#_439a0ks1gs7t)

[main.js 13](#_xl20acfbwpr)

[Agregar tarea 13](#_8wrw8pzpby3)

[Eliminar tarea 14](#_wza44395go5)

[Corrección de errores 16](#_apf5ujw1l31l)

[Vinculación de .js con .html 17](#_je03ycytsckz)

[**Primera comprobación 18**](#_thf66abzbp4i)

[Agregar tarea 18](#_da29djsrtirw)

[**II. Proceso de ToDoList 20**](#_3k7lf37wo91e)

[Opcional: Agregar fecha 20](#_jogkndyqf94h)

[**Segunda comprobación 23**](#_ctb16i7bfa21)

[Calendario, fecha, date 23](#_laqnsj2il3y9)

[**III. Proceso de ToDoList 24**](#_2v7ccho40vu4)

[Marcar tarea como completada 24](#_afxainrhg4dy)

[**Tercera comprobación 26**](#_k1n4pwxixaj9)

[Marcar tarea como completada 26](#_eyk6tjb64fcm)

[**Actualizamos repositorio 27**](#_klrd3468dr1u)

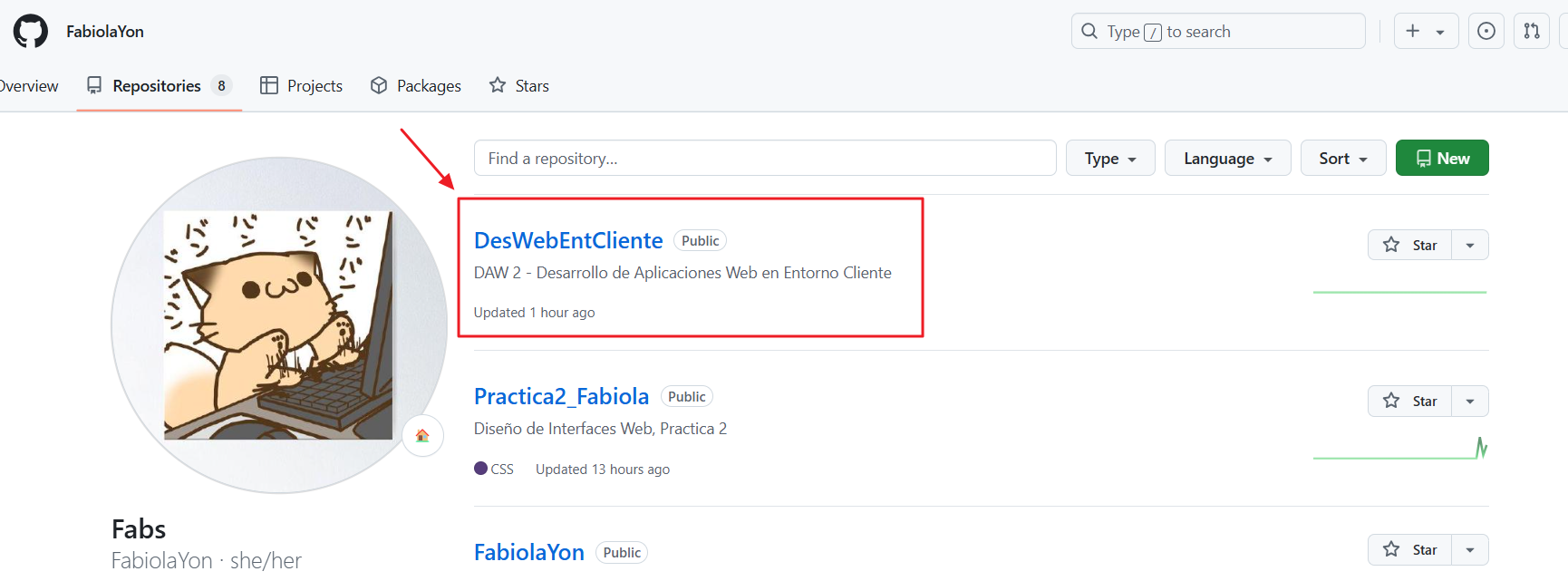
FABIOLA AYON PEREZ

# Documentación sobre GitHub

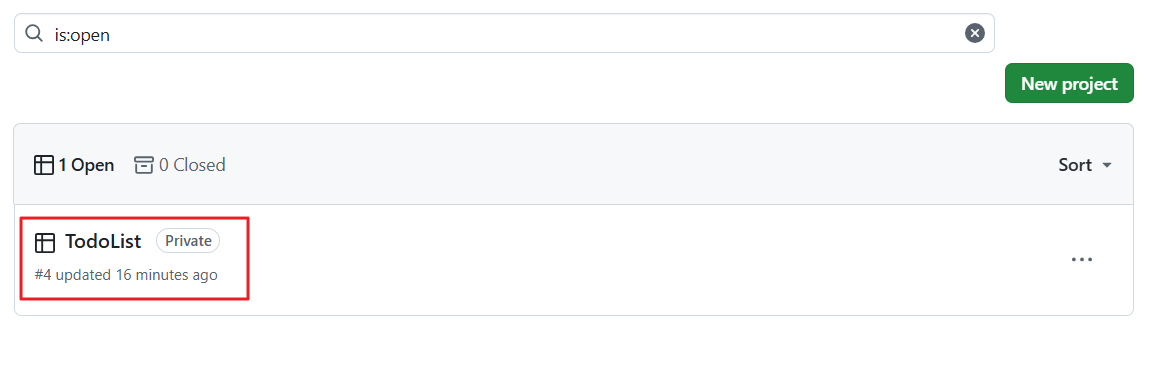
## Creación de Repository y Proyect en GitHub

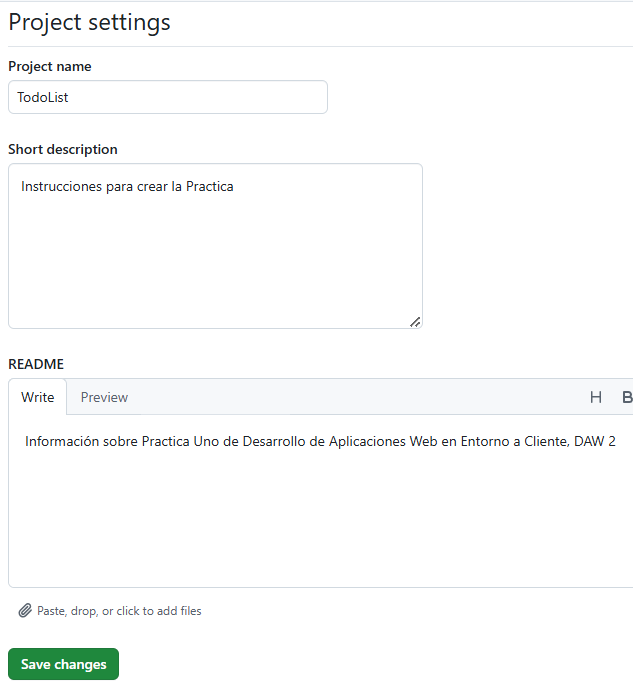
He creado un repositorio en GitHub:

<https://github.com/FabiolaYon/DesWebEntCliente.git>

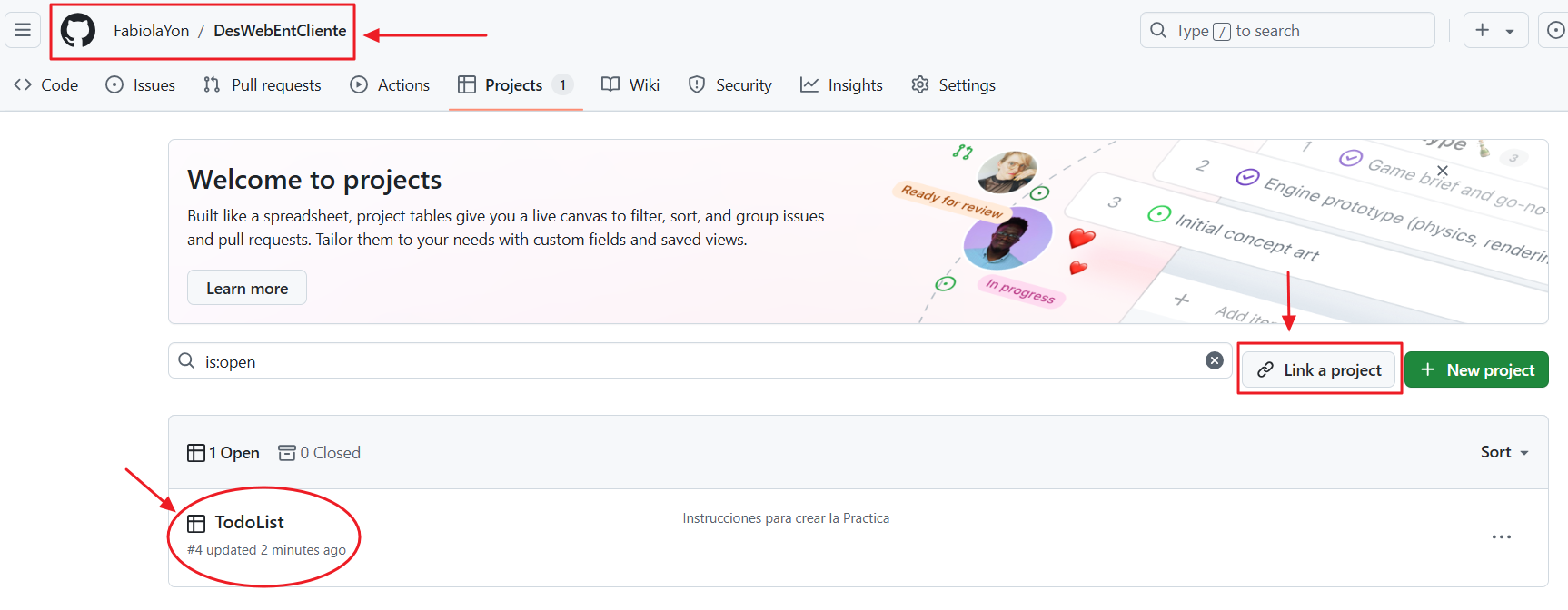


Luego he creado un proyecto TodoList



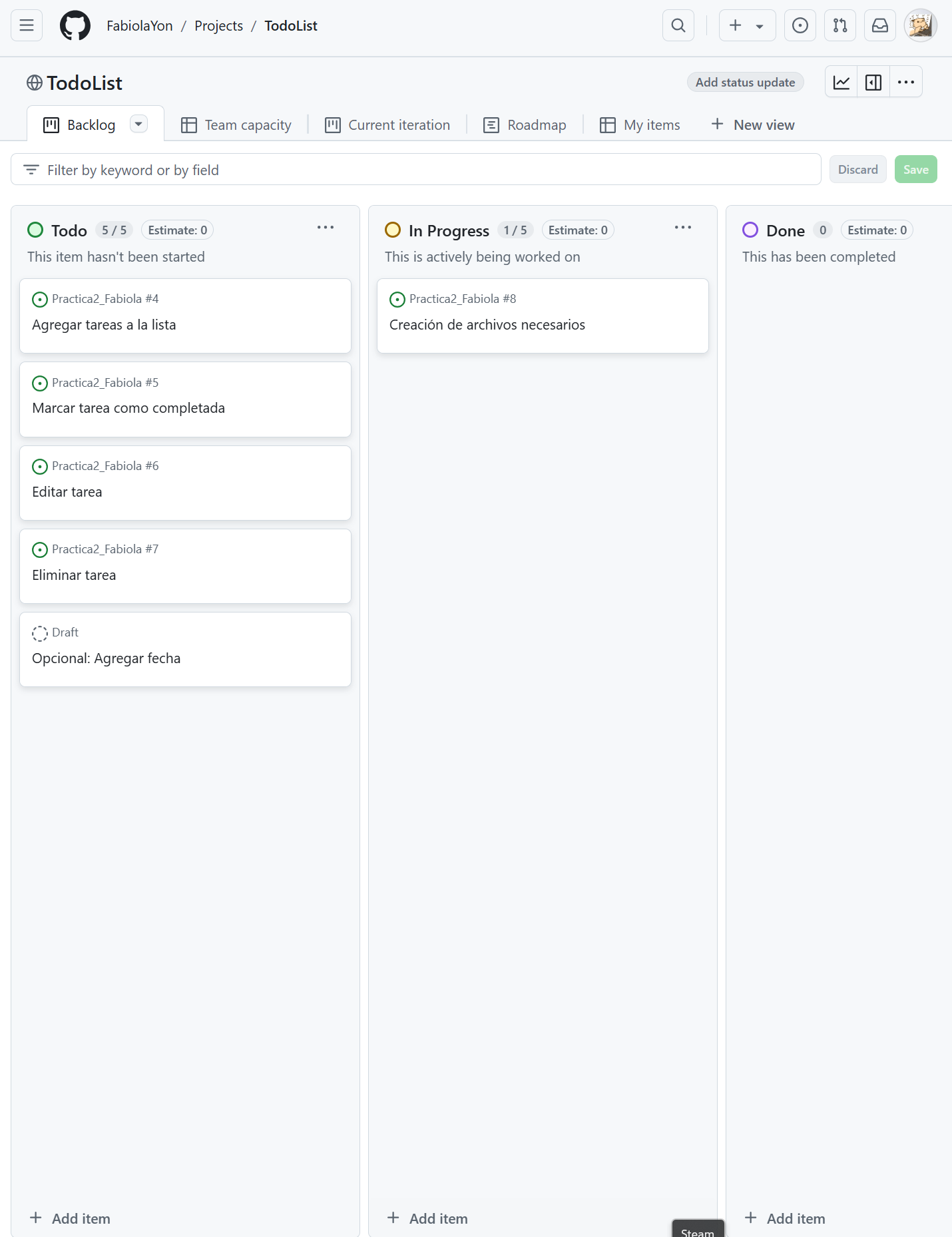


Dentro de mi repositorio he ido a la sección de Projects y he vinculado el Proyecto creado



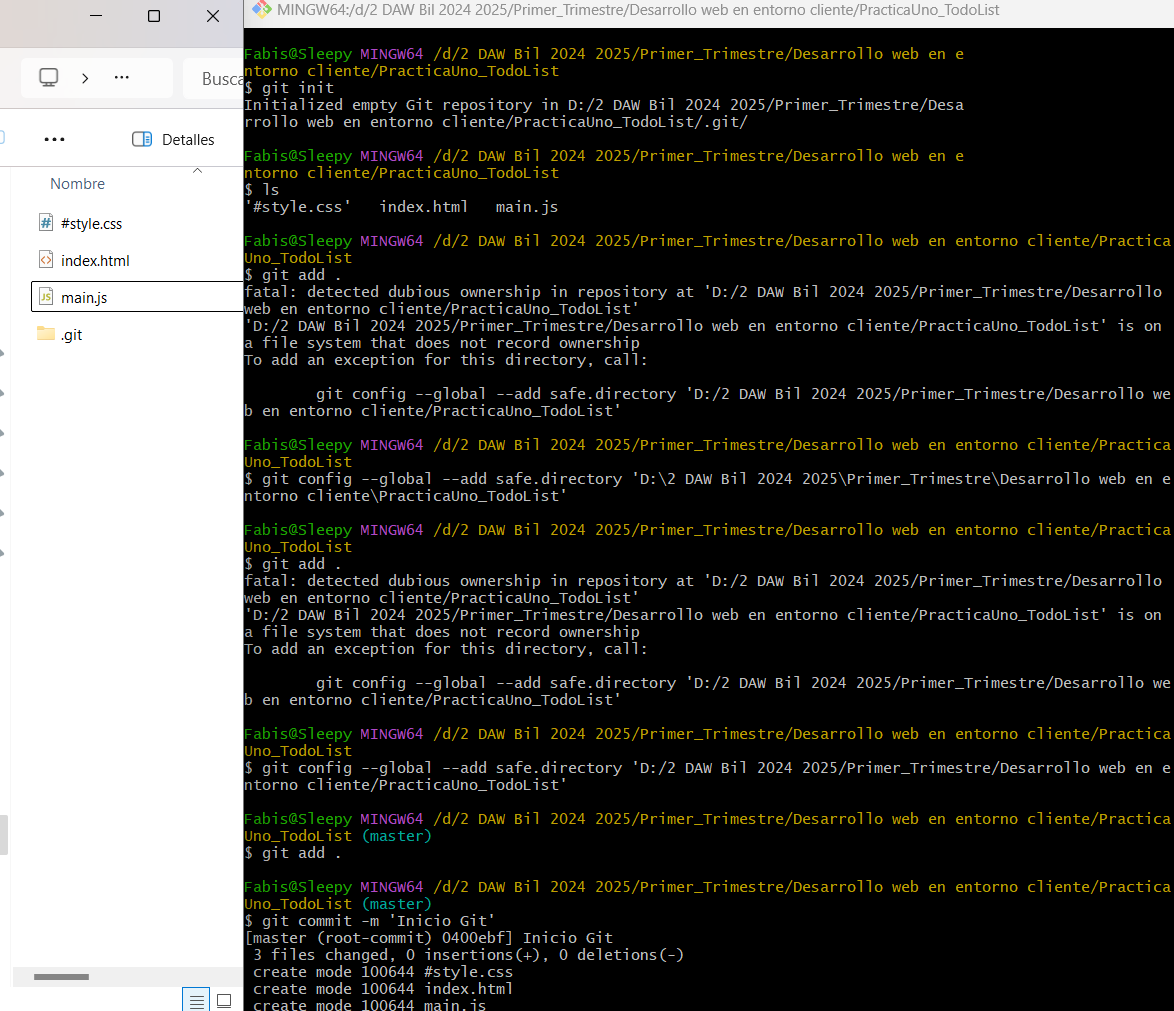
## Tareas en Proyect

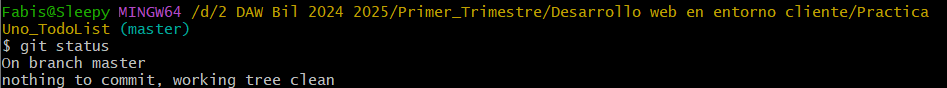
Nos ponemos en In Progress para ir creando los archivos y vincular a GitHub



## Creación de archivos y vinculación a GitHub

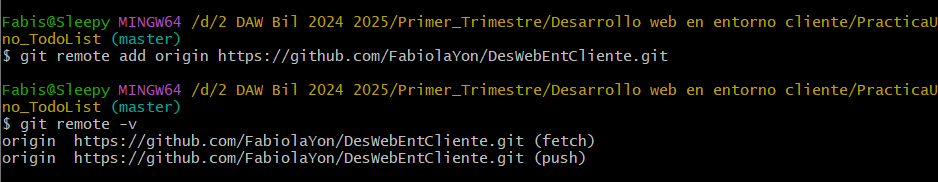
### Iniciamos Git



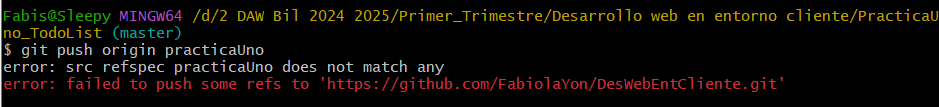


### 

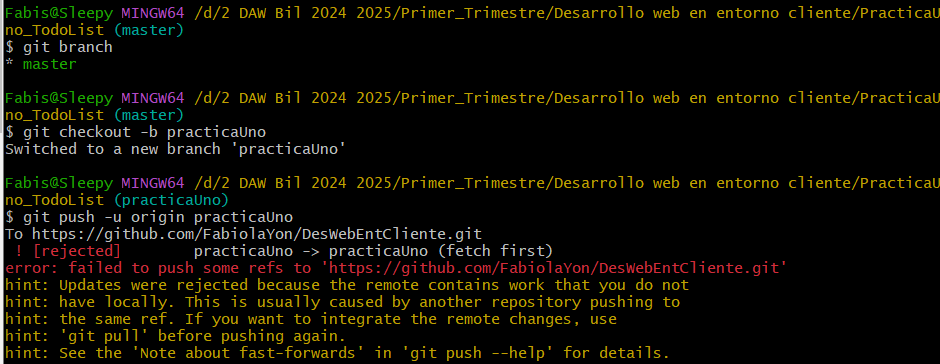
### Vinculación de archivos al repositorio



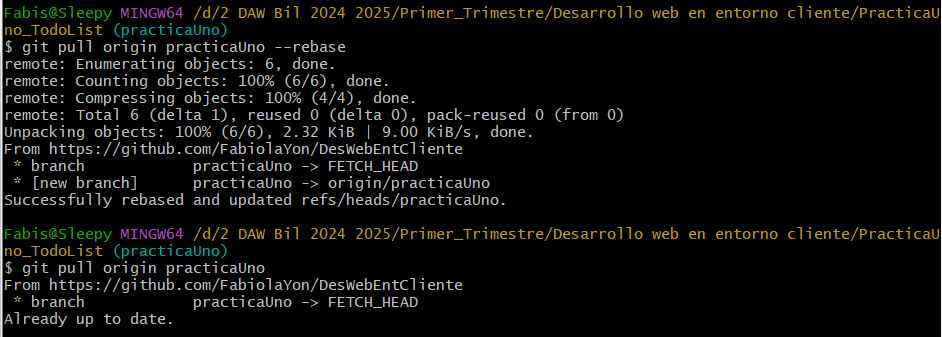
Aquí nos indica que no existe el branch de practicaUno



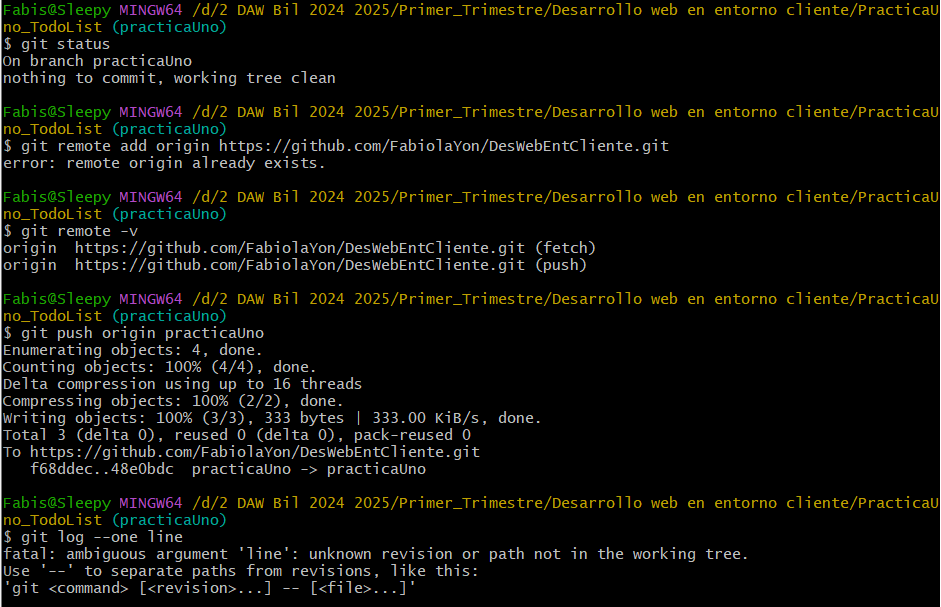
Comprobamos de que es así y creamos el branch, puede que exista en local porque lo creamos anteriormente pero no en remoto, o al revés



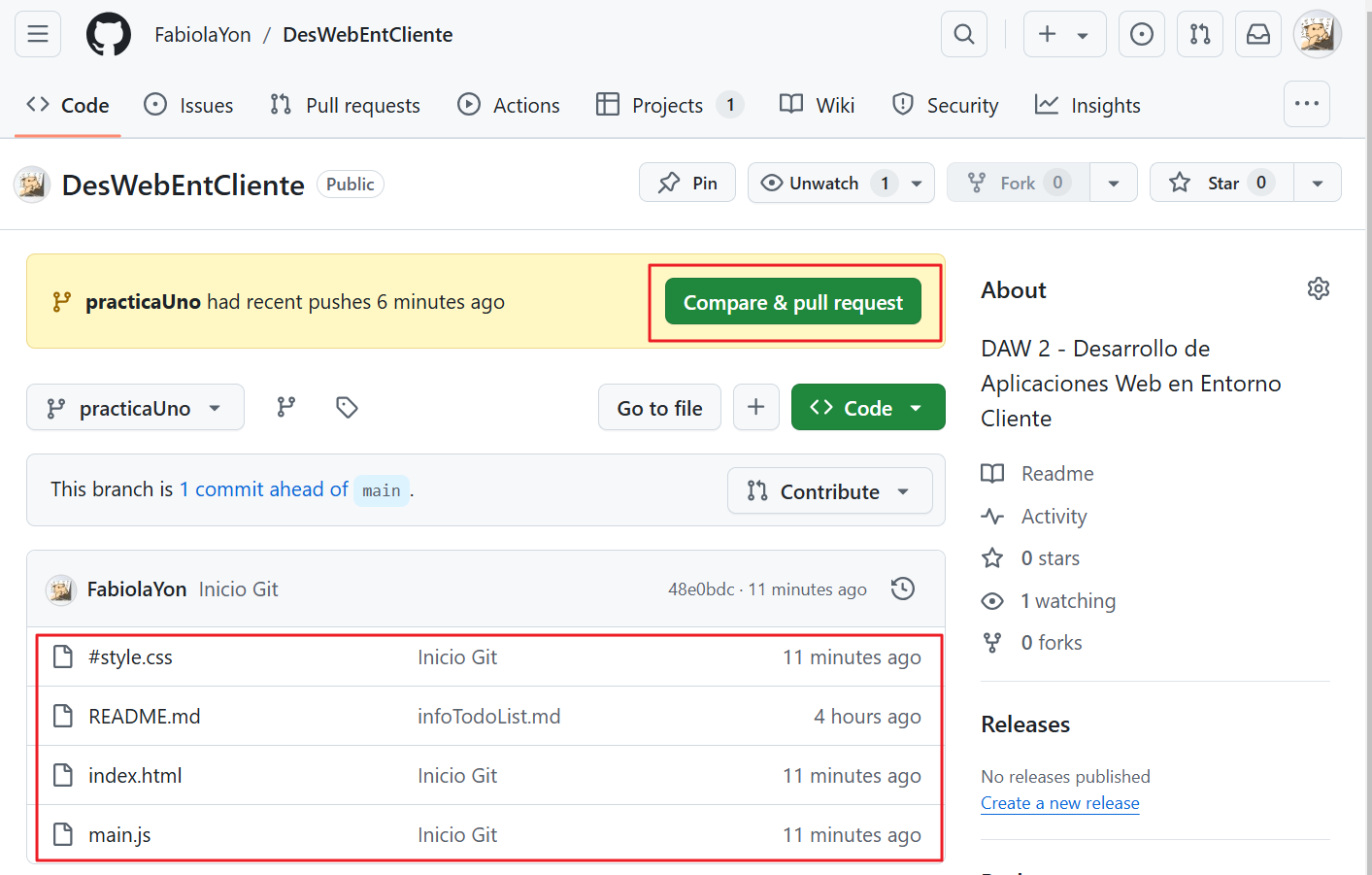
Evitamos conflictos sincronizando los cambios remotos con la rama local

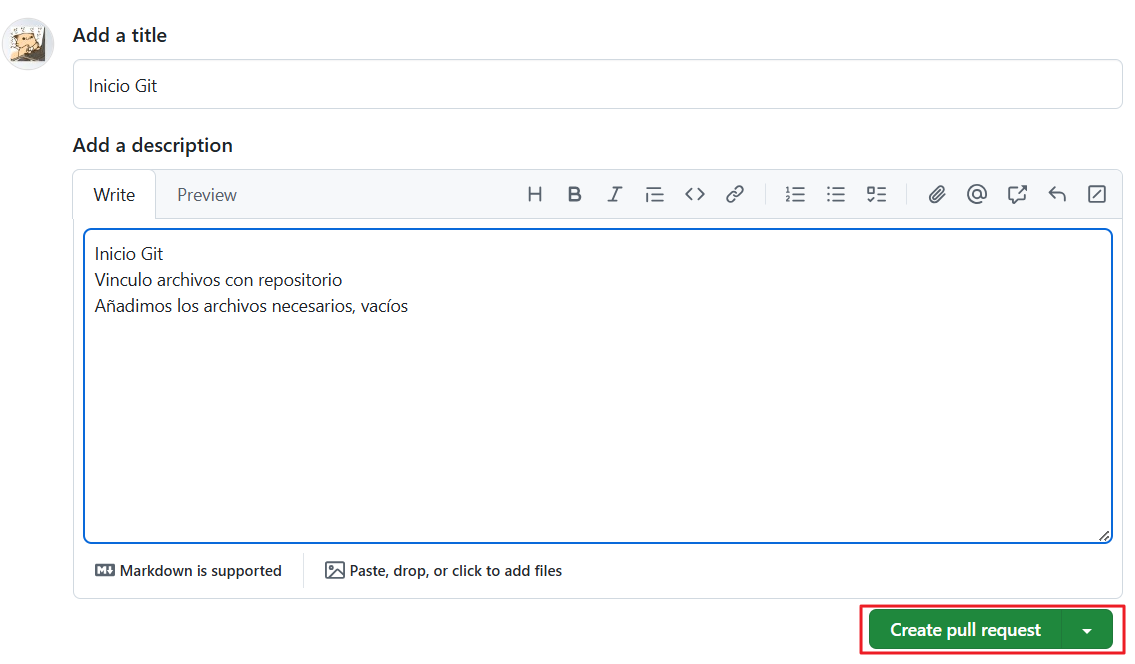


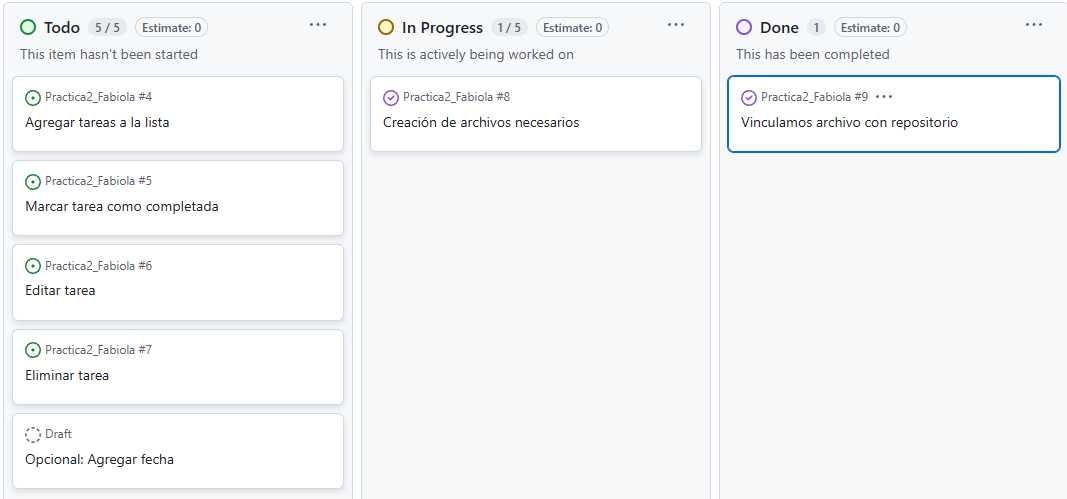
Tras lo anterior compruebo que todo esté ajustado correctamente y actualizo los archivos.



Actualizo la página de Github y aparecen ya nuestros archivos y aceptaremos los commits de practicaUno

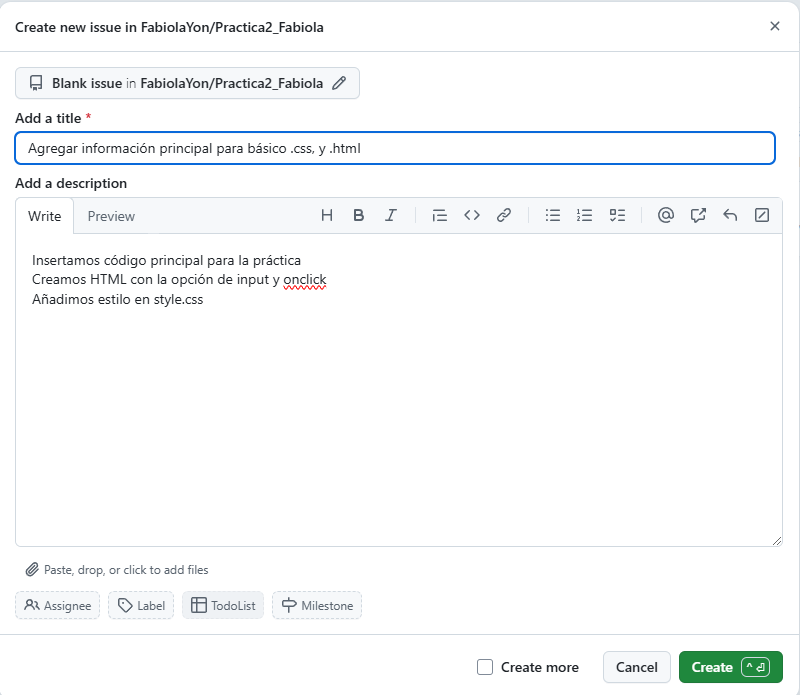


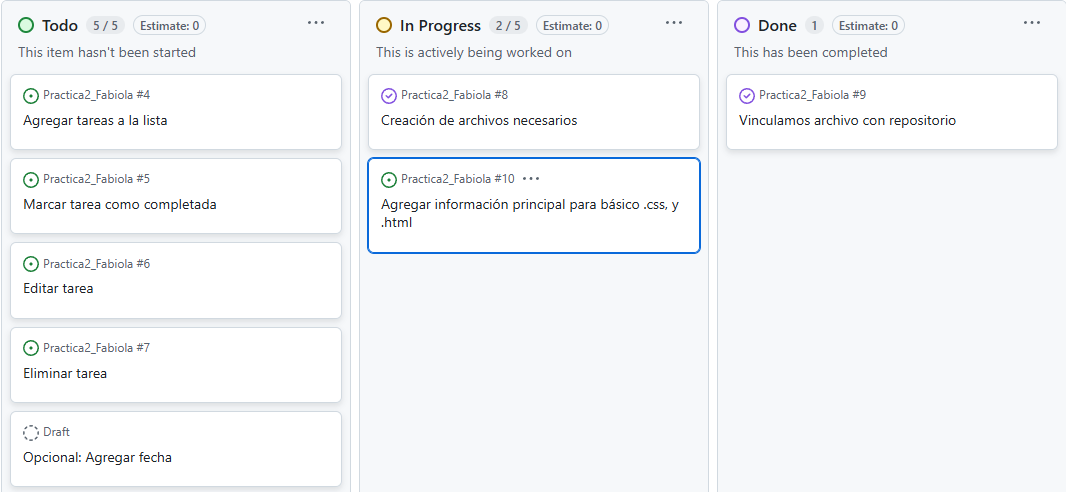




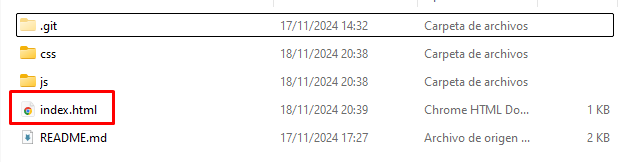
# I. Proceso de ToDoList

He creado una carpeta que contendrá los estilos, otra java script y por último mi html. En el caso de README.md es solo informativo. Iré colocando una imagen del código y luego una imagen de cómo se ve en navegador.

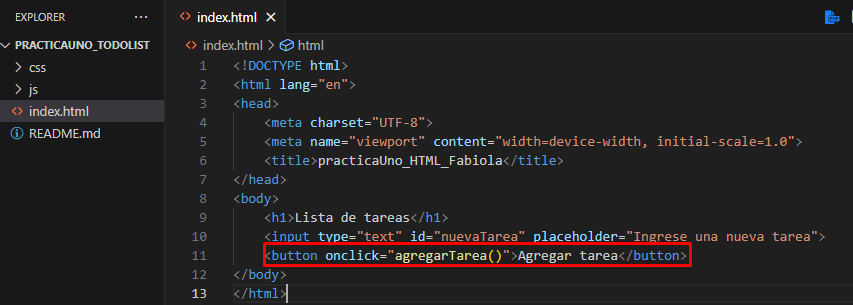


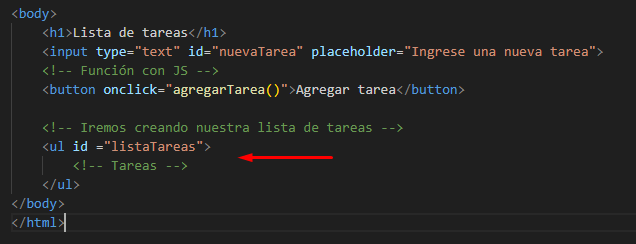


## index.html



Inicio con HTML donde colocaremos un título, un input donde agregaremos texto y luego una función de java script que crearemos más adelante.

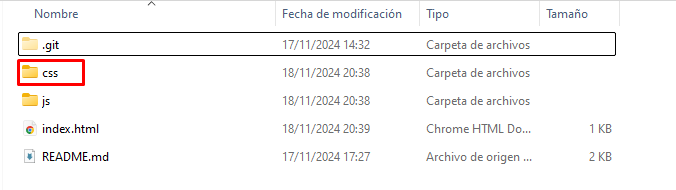






Pasaremos a darle estilo con CSS.

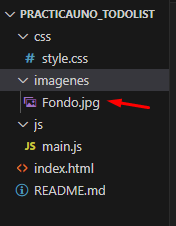
## style.css





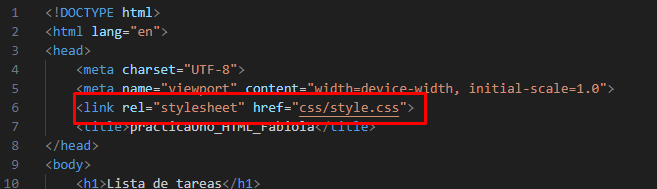
Con CSS:

* Cambiaremos el estilo de letra
* Cambiaremos el estilo de fondo, agregaré una carpeta llamada imágenes que contendrá las imágenes que utilizaré para .css





Para poder aplicarlo vamos a vincularlo con el .html en <head>



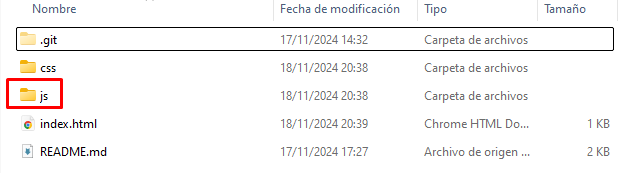
Si actualizamos la página del navegador



y continuamos con java script.

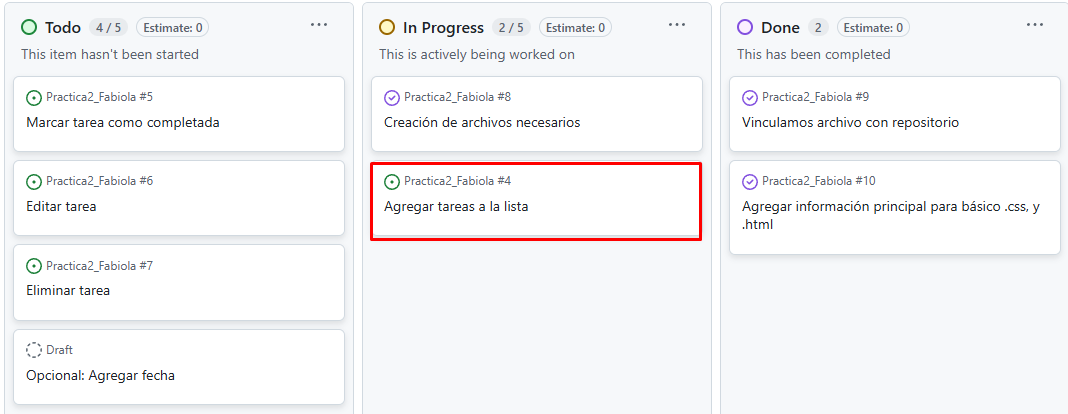
## 

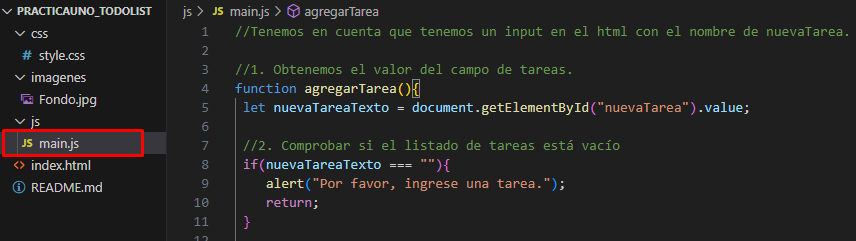
## main.js



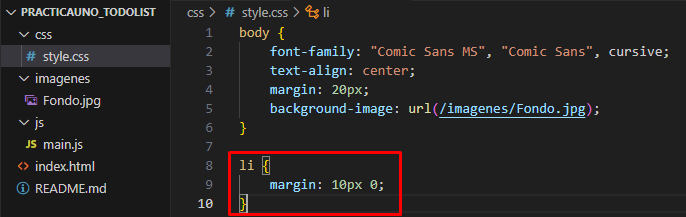
### Agregar tarea



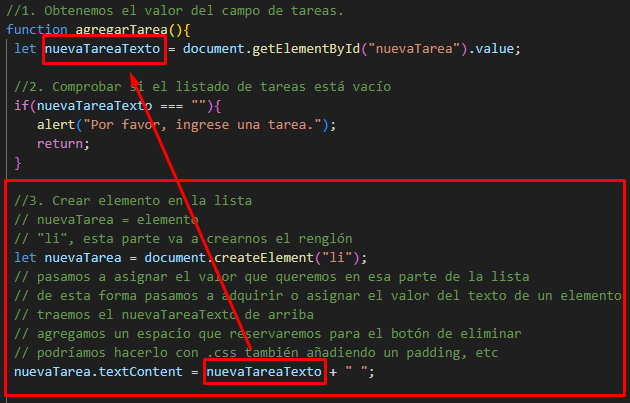




Hemos puesto que los elementos de nuestra lista de tareas estará compuesto por <li>

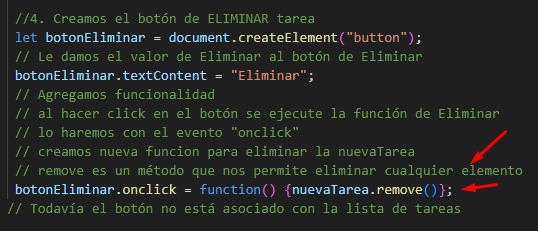


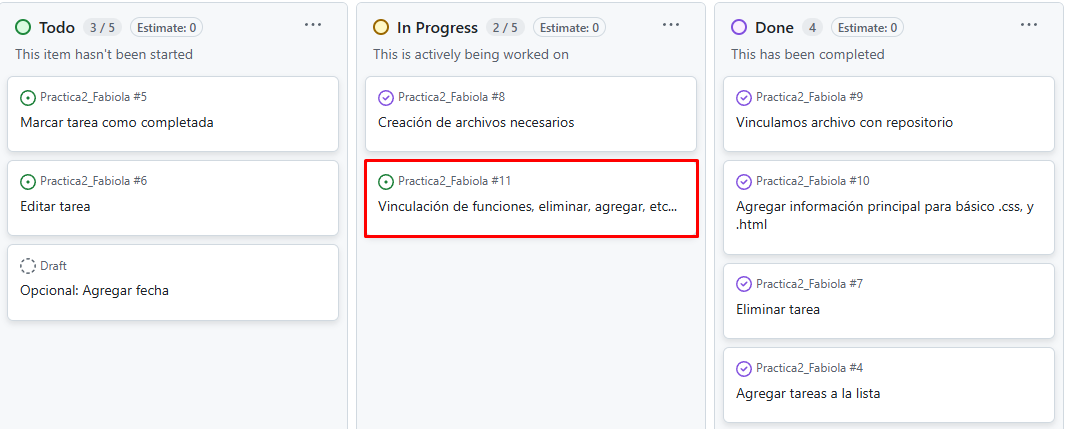
Entonces utilizaremos el método:

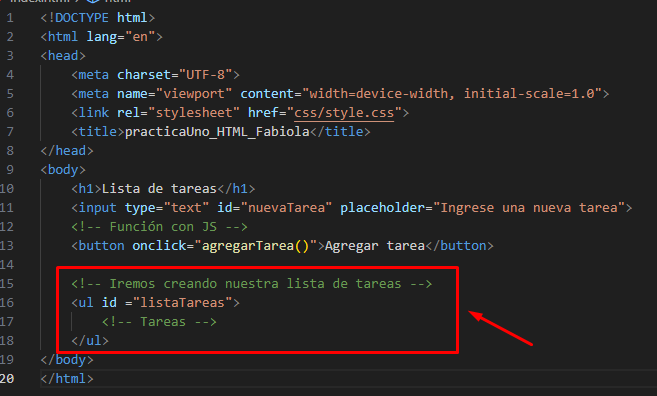


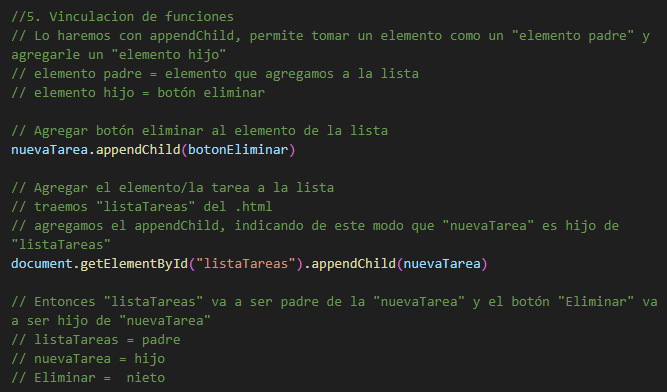
### Eliminar tarea





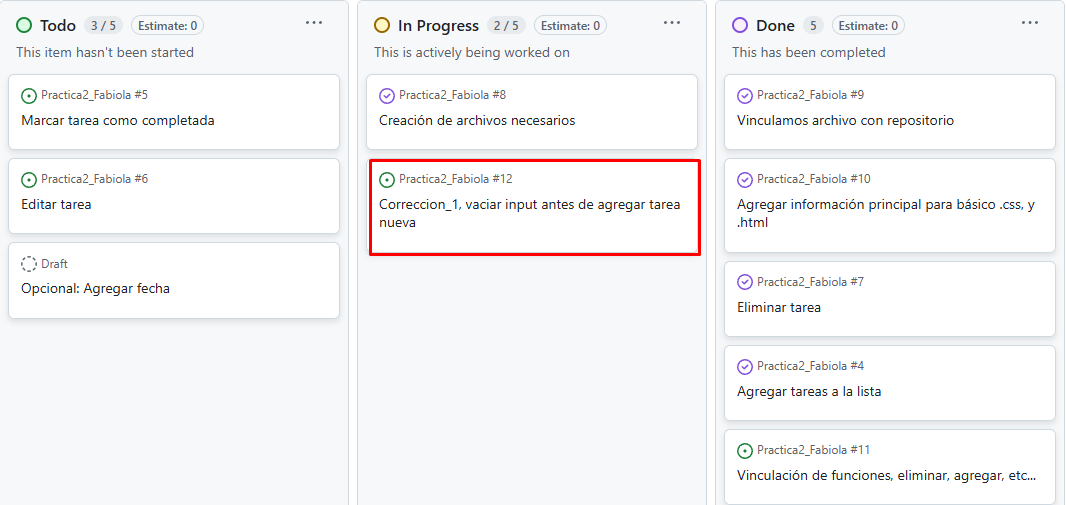


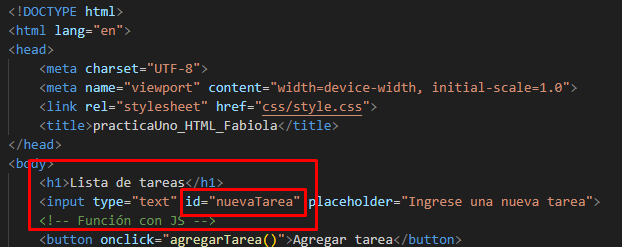


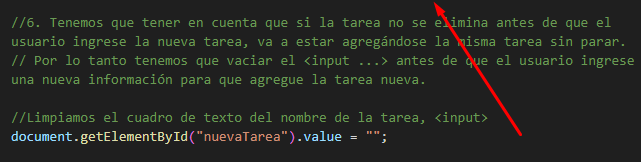


### Corrección de errores

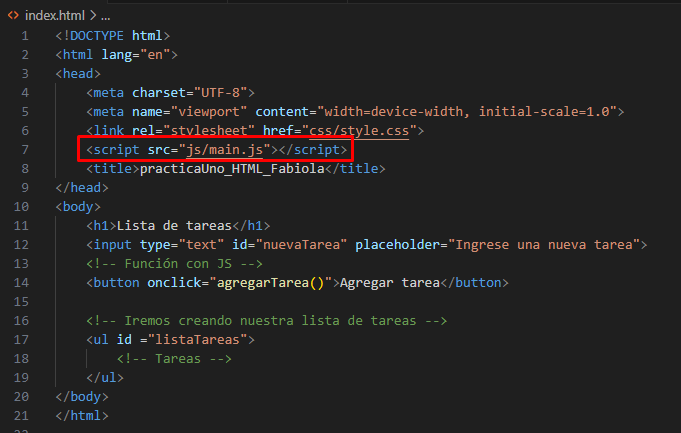
1. Vaciar el input antes de agregar tarea nueva







### Vinculación de .js con .html

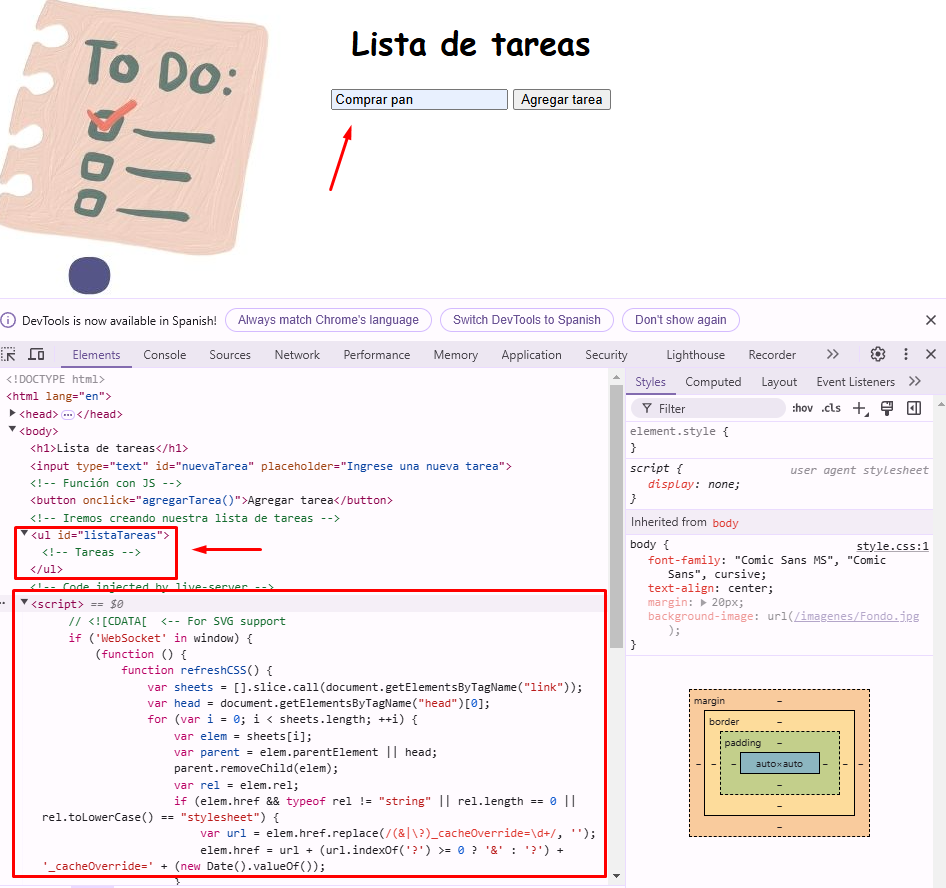


# Primera comprobación

## Agregar tarea

Apreciamos que no tenemos ninguna tarea agregada.

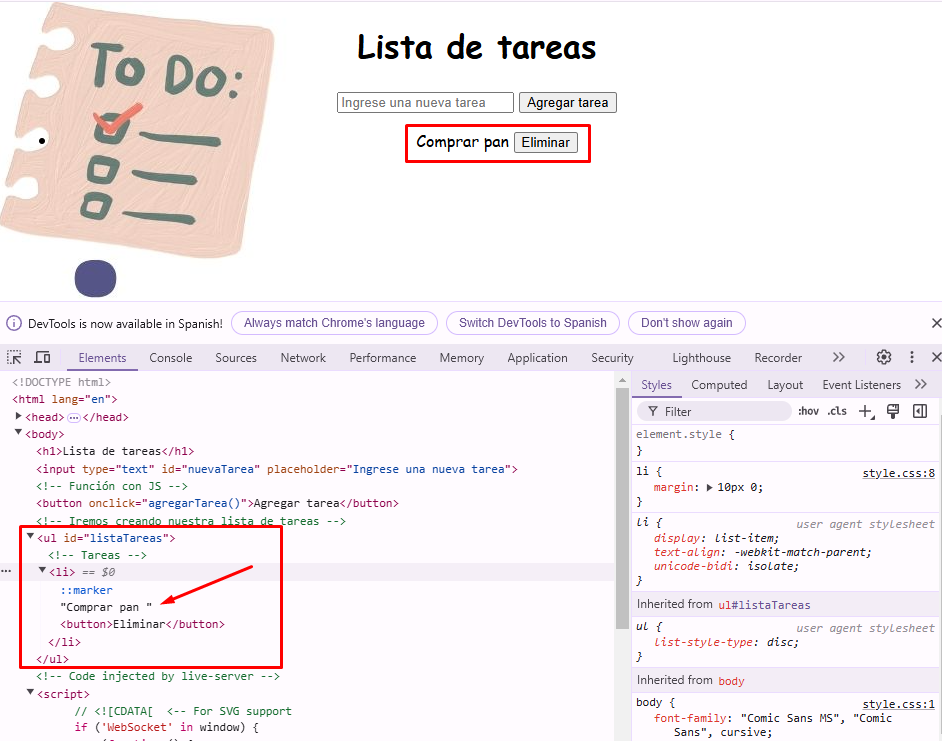
Inseramos “Comprar pan” y click en Agregar tarea.



Ahora sí, aparece la tarea junto al botón de eliminar.

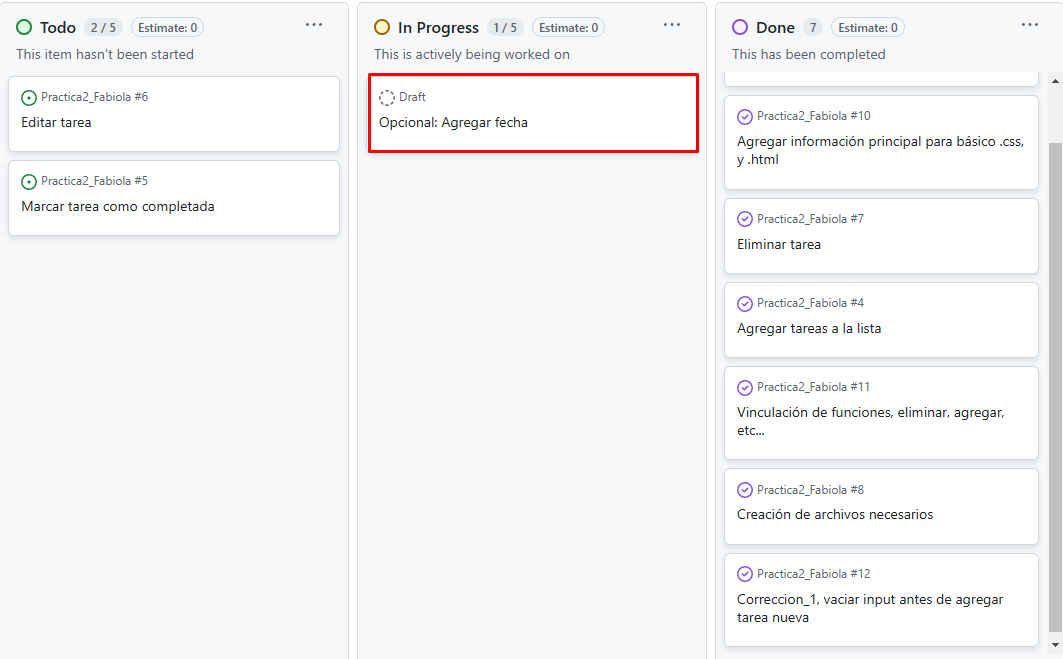
Al hacer click en eliminar la tarea desaparece.

Volvemos a estar como al principio y por lo tanto comprobamos que funciona.

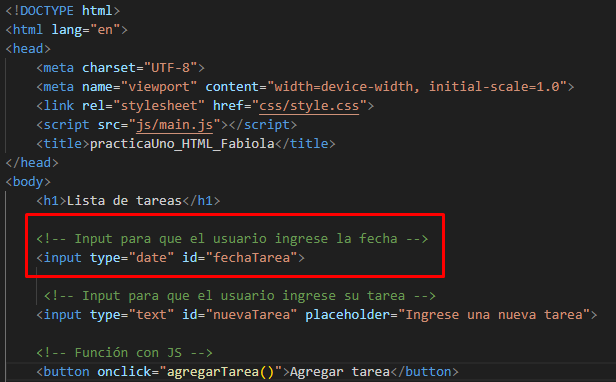


# II. Proceso de ToDoList

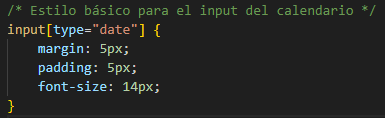
## Opcional: Agregar fecha



Creamos un input para la fecha



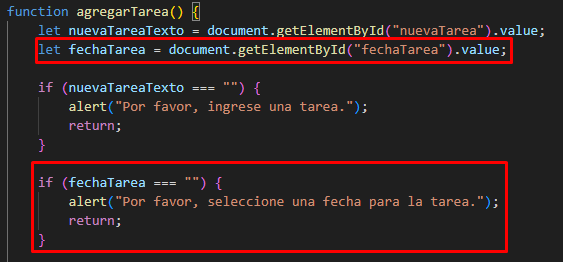
Le daremos un estilo con css para que luzca como un calendario. También asignaremos tamaño.



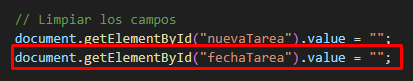
Como anteriormente hicimos con las tareas.

Tomamos el valor del .html

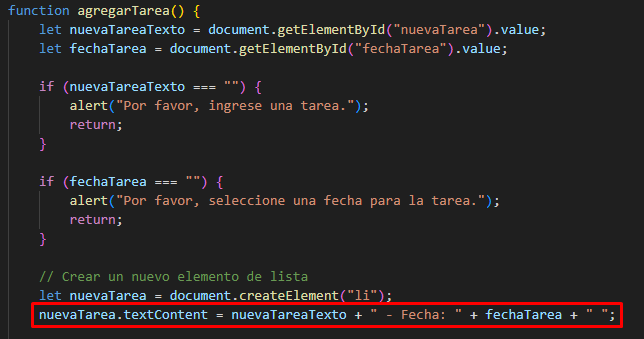
Creamos alert por si no se ha seleccionado una fecha



También limpiaremos los campos de la fecha para que no hayan errores de bucle



Agregamos la opción de fecha y volvemos a dejar un espacio para el botón de Eliminar

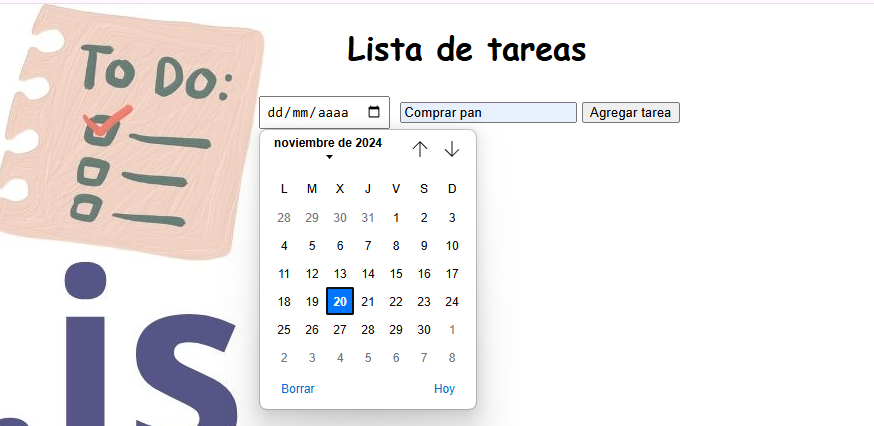


Comprobamos si esto funciona

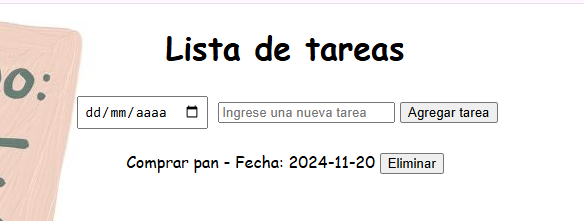
# 

# Segunda comprobación

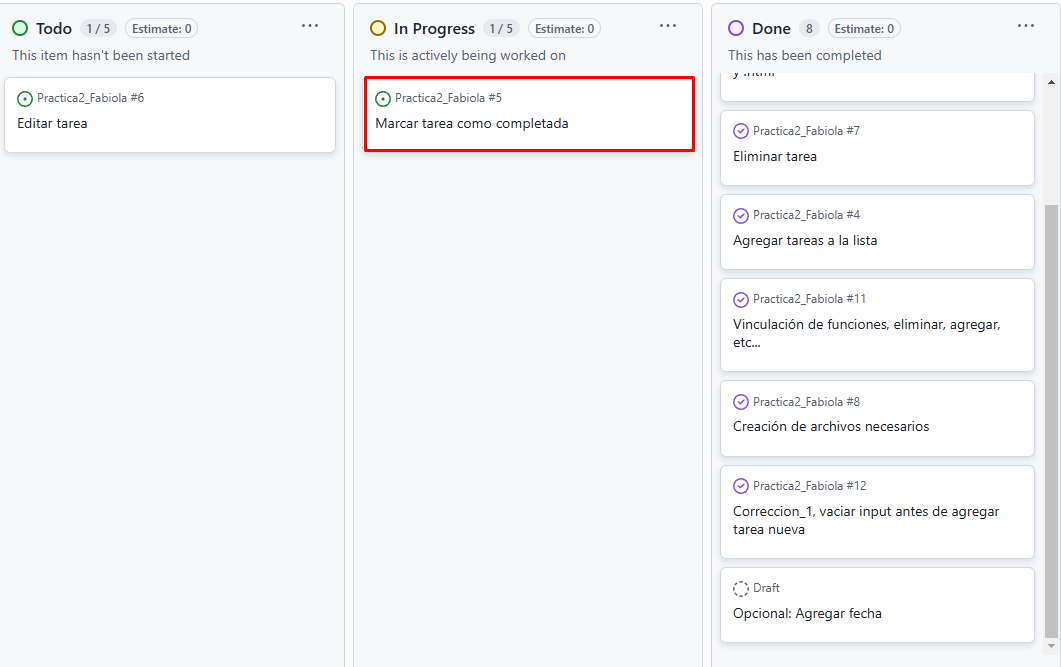
## Calendario, fecha, date



Aparece el calendario y la fecha al lado de la tarea



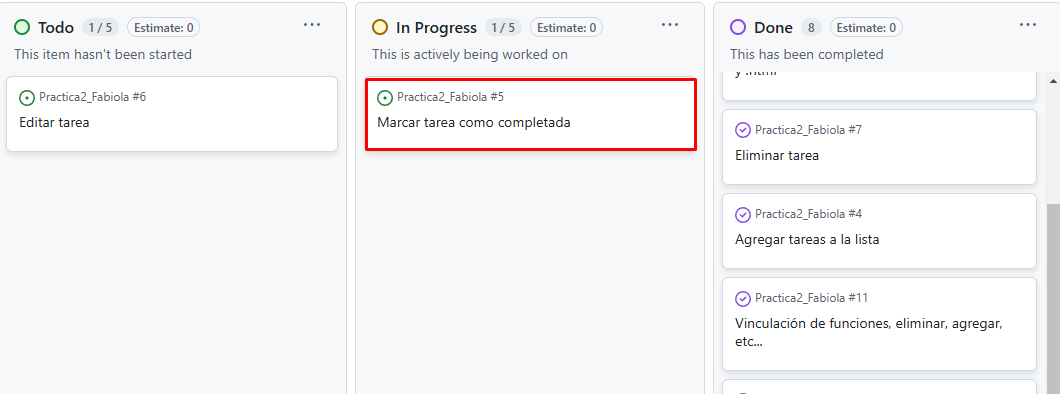
Marco en proyecto de Git tarea como completada.



# III. Proceso de ToDoList

## Marcar tarea como completada

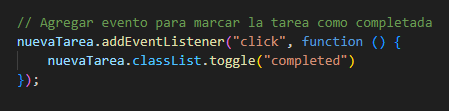
En este caso haré que el color de letra cambie y se tache. Esto se hará con css.



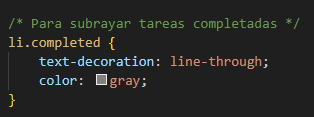
Para ello en JS creamos un nuevo evento que marque la nuevaTarea como “completed” de <li>

La nuevaTarea tiene la etiqueta <li>





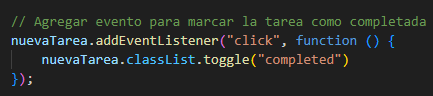
En CSS



* li.completed: Selecciona los elementos <li> que tengan la clase completed.
* text-decoration: line-through;: Subraya el texto con una línea horizontal (efecto de tachado).
* color: gray;: Cambia el color del texto a gris para que visualmente parezca que está "hecho".

Cuando el usuario haga clic, la clase completed se aplica o se quita, y los estilos se activan o desactivan según corresponda.

Volviendo al .js



Usaremos toggle que es un método de JavaScript que se usa para agregar o eliminar una clase.

* Si no tiene la clase completed, el método toggle la agrega, y el CSS tacha la tarea y la pone en gris.
* Si ya tiene la clase completed, el método toggle la elimina, devolviendo la tarea a su estado original.

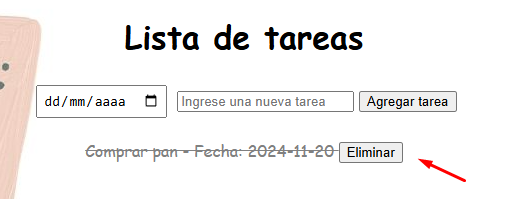
Entonces…

* Cuando el usuario hace clic en una tarea, el navegador detecta el evento gracias a addEventListener.
* La función cambia las clases del elemento <li> con toggle.
* El CSS aplica o elimina los estilos de la clase completed, dando el efecto de marcar o desmarcar como completado.

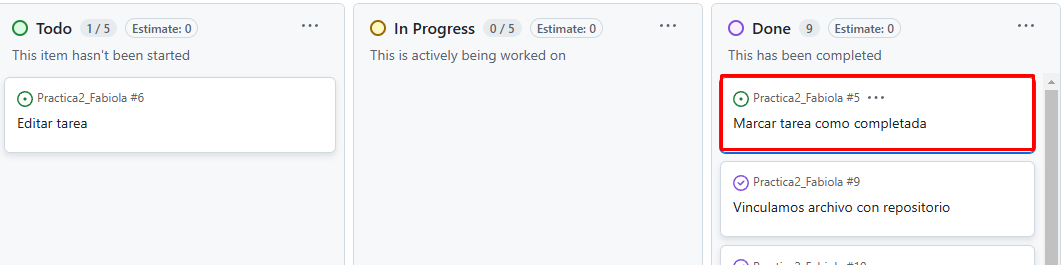
Comprobamos en navegador.

# Tercera comprobación

## Marcar tarea como completada



Funciona correctamente.



La parte de editar tarea la dejaré para más adelante. 🙂

# Actualizamos repositorio

