

Tarea2 - Ejercicio de Git - proyecto labdist

Adrián García de la Cera

Enlace repositorio GitHub:

<https://github.com/fuejomusta/labdistAdri-nTarea2.git>

Enlace video presentación:

Tarea2 - Ejercicio de Git - proyecto labdist

Requisitos previos

Trabajo en local

- 1.Inicializa un nuevo repositorio Git en una carpeta llamada "labdist" y agrega los archivos proporcionados en el aula virtual. Renombra la rama master a main , si es necesario. Realiza el primer commit. Muestra el log del repositorio.
- 2.Incluye un fichero .gitignore para que los ficheros README.md , LICENCE.txt y passwords.txt sean ignorados por el control de versiones. Realiza el commit y muestra los logs del repositorio en una línea.
- 3.En el repositorio, crea los archivos README.md , LICENCE.txt y passwords.txt con algún contenido. Muestra el estado del repositorio. Muestra el listado de archivos ignorados.
- 4.Crea una rama feature-estilos . Cámbiate a ella.
 - Modifica el archivo estilos.css :
 - propiedad color del body y de h2 : #2a2a2a
 - propiedad background-color de header y footer: #2a75ff
 - Comprueba el estado del repositorio. Añade los cambios, realiza un commit con el mensaje "actualizados estilos a azules"
- 5.Vuelve a la rama main . En el archivo index.html añade un comentario donde se indique tu nombre como autor de la página. Comprueba el estado del repositorio. Añade los cambios, realiza un commit con el mensaje 'añadido autor en index'. Muestra los logs del repositorio en una línea, gráficamente y con 'decoración'
- 6.Fusiona la rama feature-estilos en la rama main . Muestra los logs del repositorio en una línea, gráficamente y con 'decoración'

Trabajo en remoto

- 1.Continúa con el repositorio labdist . Añade el repositorio a Sourcetree.
- 2.En tu cuenta de GitHub, crea un repositorio remoto y sube al remoto los ficheros de tu repositorio local. Debes subir todas las ramas. Muestra, además, la captura de pantalla donde se vean en GitHub.
- 3.En el repositorio local, crea una rama feature-index . Añade el siguiente código dentro de la <section class="about"> . Añade los cambios y crea un commit con el mensaje "Añadido párrafo equipo en index.html". Sube los cambios al remoto. (Recuerda que debes usar SourceTree en todo este apartado)
- 4.En el repositorio local, fusiona la rama feature-index en la rama main .
- 5.Edita el fichero contacto.html. Borra unas líneas. Muestra los ficheros con cambios pendientes y las diferencias. Añade los cambios y haz un commit.
- 6.Te das cuenta del error. Deshaz TOTALMENTE el commit anterior. Captura el estado actual del repositorio. (Asegúrate de que el fichero contacto.html ha recuperado todas las líneas borradas y no hay cambios pendientes en el repositorio.)
- 8.Crea una rama feature-mapa y cámbiate a ella. Incluye este código en el archivo contacto.html . Añade los cambios. Realiza un commit.
- 8.Sube los cambios al remoto - los de todas las ramas. Muestra en el remoto los cambios del archivo contacto.html en la rama feature-mapa .

9. En GitHub, en la rama `main`, fusiona la rama `feature-mapa`. Baja los cambios del remoto a local. Deja los dos repositorios sincronizados. Muestra una captura de pantalla donde se vea la página principal de tu repositorio remoto

Conflictos

1. Crea una rama `hotfix-js`. Cámbiate a ella. Añade este código en el fichero `script.js`. Confirma el cambio y haz un commit con el mensaje "corregido problema en `script.js`".
2. Vuelve a la rama `main`. En el fichero `script.js` en las mismas líneas que en la cuestión anterior, añade el código siguiente. Confirma el cambio y haz un commit con el mensaje "corregido problema en `script.js` rama `main`".
3. Fusiona la rama `hotfix-js` en `main`. Debe producirse un conflicto. Resuélvelo como consideres oportuno. Cuando termines la resolución del conflicto sube los cambios al remoto.

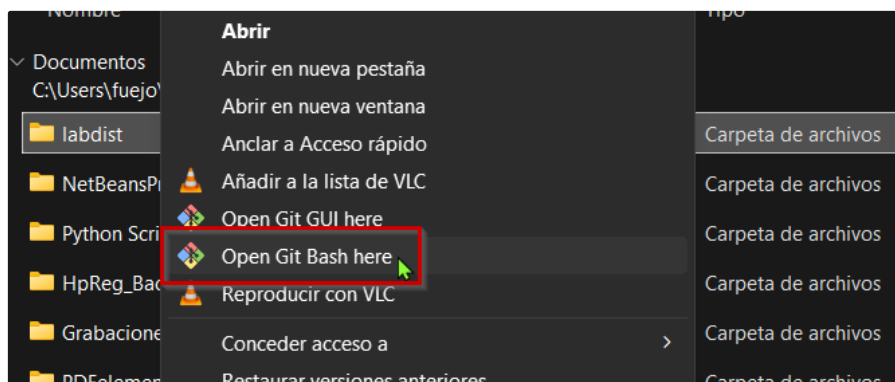
Requisitos previos

Tener cuenta de GitHub

Tener instalado Sourcetree

Trabajo en local

Al trabajar en local, creo la carpeta `labdist` con el entorno gráfico, realizo clic derecho sobre ella y selecciono `Open Git Bash here` (para crear la carpeta a través de comandos usaríamos el comando `mkdir labdist`).



Una vez abierto, nos abre la consola de comandos de Git, introducimos el comando `git config --global merge.ff false` para que el historial de ramas se mantenga como árbol.

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist
$ git config --global merge.ff false
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist
$
```

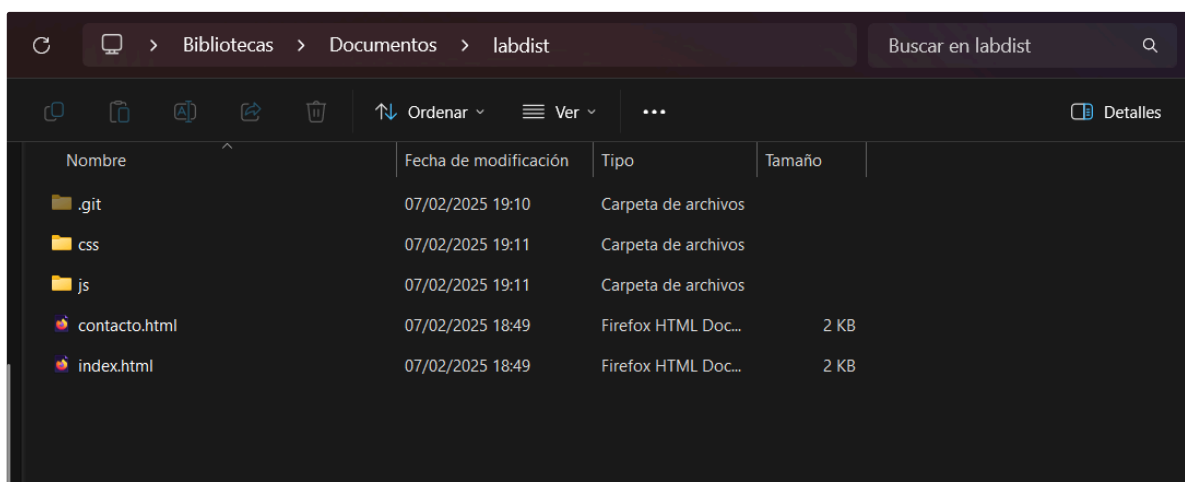
1. Inicializa un nuevo repositorio Git en una carpeta llamada "labdist" y agrega los archivos proporcionados en el aula virtual. Renombra la rama master a **main** , si es necesario. Realiza el primer **commit**. Muestra el log del repositorio.

Para inicializar el repositorio en la carpeta usamos el comando:

```
git init
```

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist
$ git config --global merge.ff false
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist
$ git init
Initialized empty Git repository in C:/Users/fuejo/OneDrive/Documentos/labdist/.git/
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (master)
$
```

Una vez iniciado copiamos en la carpeta los archivos del aula virtual.



Con el comando **git status** vemos los comandos que ya detecta los archivos introducidos.

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
$ git config --global merge.ff false

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist
$ git init
Initialized empty Git repository in C:/Users/fuejo/OneDrive/Documentos/labdist/.git/

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    contacto.html
    css/
    index.html
    js/

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

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (master)
$ |
```

Renombramos la rama master a main con el comando:

```
git branch -m master main
```

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
    index.html
    js/

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

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (master)
$ git branch -m master main

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    contacto.html
    css/
    index.html
    js/

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

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ |
```

Usamos el comando `git add .` para guardar los cambios en el área de preparación y con `git status` comprobamos que están subidos.

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
    index.html
    js/

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

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git add .

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status
On branch main

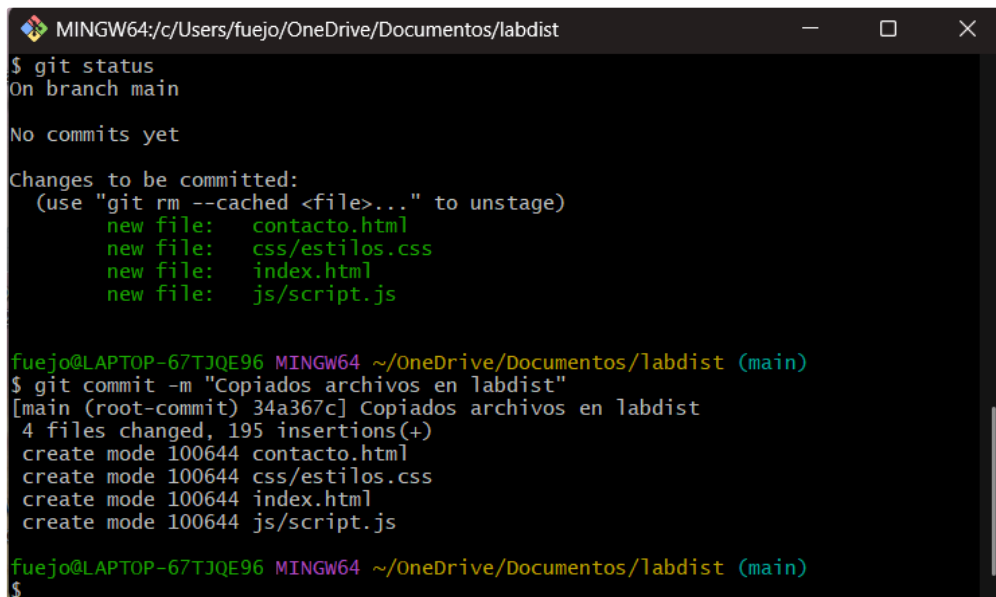
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   contacto.html
    new file:   css/estilos.css
    new file:   index.html
    new file:   js/script.js

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ |
```

Una vez en el área de preparación, se realiza el **commit** al que nombraremos "Copiados archivos en labdist":

```
git commit -m "Copiados archivos en labdist"
```



```
MINGW64:/c:/Users/fuejo/OneDrive/Documentos/labdist
$ git status
On branch main

No commits yet

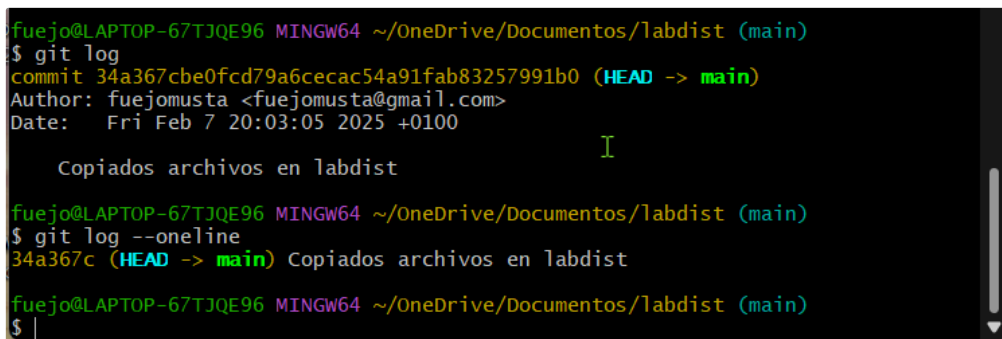
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   contacto.html
        new file:   css/estilos.css
        new file:   index.html
        new file:   js/script.js

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git commit -m "Copiados archivos en labdist"
[main (root-commit) 34a367c] Copiados archivos en labdist
 4 files changed, 195 insertions(+)
 create mode 100644 contacto.html
 create mode 100644 css/estilos.css
 create mode 100644 index.html
 create mode 100644 js/script.js

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

Mostramos los log del repositorio con **git log** o con **git log --oneline** para que lo muestre solo en una línea.

```
git log --oneline
```



```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git log
commit 34a367cbe0fcd79a6cecac54a91fab83257991b0 (HEAD -> main)
Author: fuejomusta <fuejomusta@gmail.com>
Date:   Fri Feb 7 20:03:05 2025 +0100

    Copiados archivos en labdist

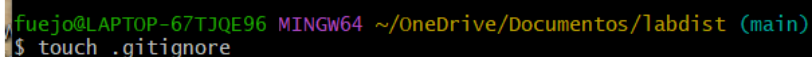
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git log --oneline
34a367c (HEAD -> main) Copiados archivos en labdist

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

2. Incluye un fichero **.gitignore** para que los ficheros **README.md** , **LICENCE.txt** y **passwords.txt** sean ignorados por el control de versiones. Realiza el commit y muestra los logs del repositorio en una línea.

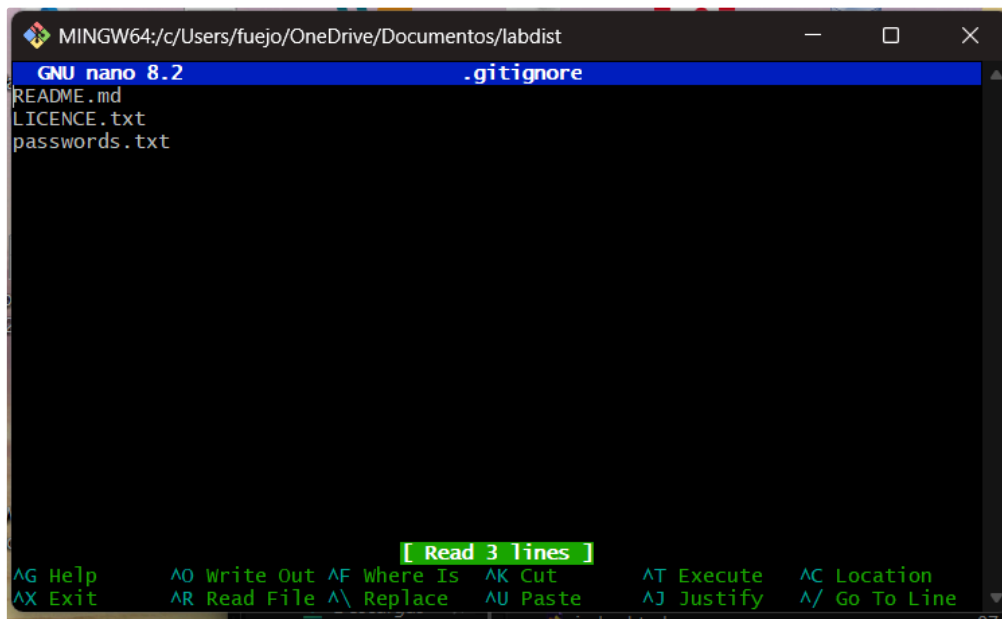
Para que Git ignore los archivos indicados, en la raíz del repositorio creamos el archivo **.gitignore** de la siguiente manera:

```
touch .gitignore
```



```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ touch .gitignore
```

Utilizo el comando **nano .gitignore** para acceder al archivo **.gitignore** e indicarle el nombre de los archivos a ignorar, también es posible abrirlo haciendo doble clic y que se abra en un IDE como Visual estudio.

A screenshot of a terminal window running the GNU nano 8.2 editor. The window title is 'MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist'. The editor is editing a file named '.gitignore'. The content of the file is: 'README.md', 'LICENCE.txt', and 'passwords.txt'. The bottom status bar shows 'Read 3 lines' and various keyboard shortcuts like 'AG Help', 'AX Exit', 'AO Write Out', etc.

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
GNU nano 8.2 .gitignore
README.md
LICENCE.txt
passwords.txt

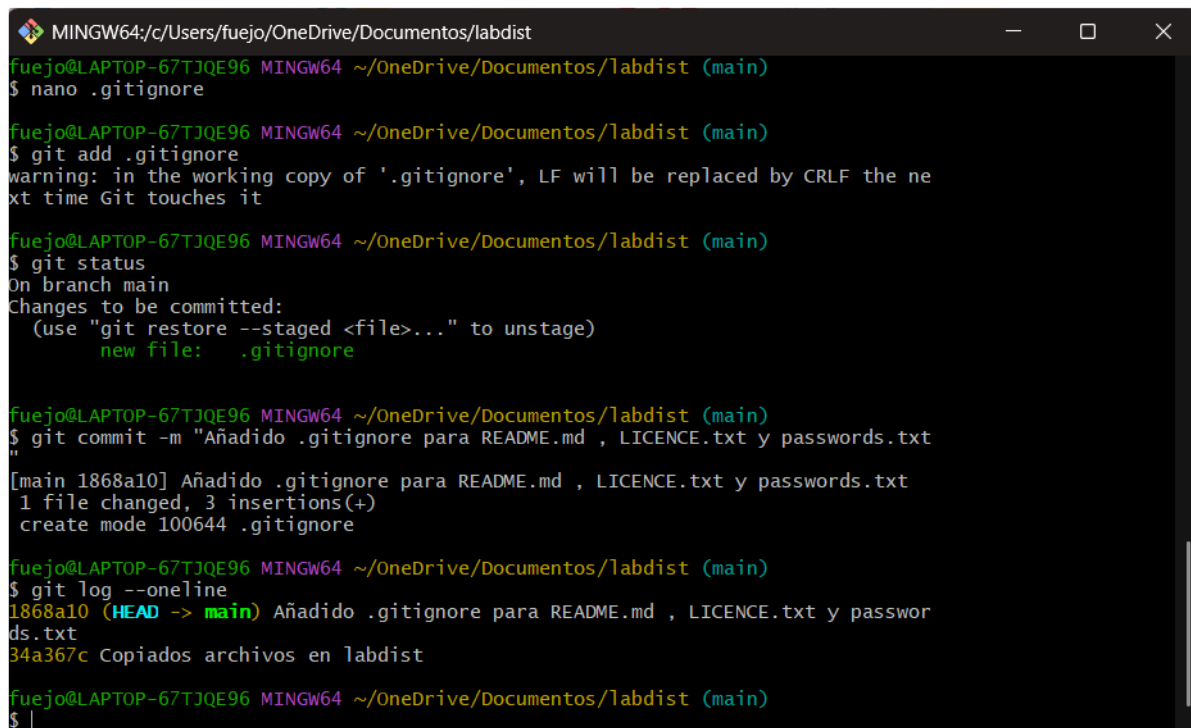
[ Read 3 lines ]
AG Help  AO Write Out AF Where Is  AK Cut      AT Execute  AC Location
AX Exit  AR Read File  AL Replace  AU Paste   AJ Justify  ^/ Go To Line
```

Subimos los cambios realizados al área de preparación y realizamos el `commit` con su nombre:

```
git commit -m "Añadido .gitignore para README.md , LICENCE.txt y passwords.txt"
```

Tras realizar el commit, se muestran los log con el comando:

```
git log --oneline
```

A screenshot of a terminal window showing a sequence of git commands and their output. The window title is 'MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist'. The user is 'fuejo@LAPTOP-67TJQE96'. The commands and output are: 'nano .gitignore', 'git add .gitignore' (with a warning about LF vs CRLF), 'git status' (showing 'new file: .gitignore'), 'git commit -m "Añadido .gitignore para README.md , LICENCE.txt y passwords.txt"', and 'git log --oneline' (showing commit 1868a10).

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ nano .gitignore

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git commit -m "Añadido .gitignore para README.md , LICENCE.txt y passwords.txt"
[main 1868a10] Añadido .gitignore para README.md , LICENCE.txt y passwords.txt
 1 file changed, 3 insertions(+)
 create mode 100644 .gitignore

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git log --oneline
1868a10 (HEAD -> main) Añadido .gitignore para README.md , LICENCE.txt y passwor
ds.txt
34a367c Copiados archivos en labdist

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ |
```

3. En el repositorio, crea los archivos **README.md** , **LICENCE.txt** y **passwords.txt** con algún contenido. Muestra el estado del repositorio. Muestra el listado de archivos ignorados.




Para crear los archivos usamos el comando **touch** seguido por el nombre del archivo (por error se ha creado el fichero **readmi** en vez de **readme**, el fichero incorrecto se ha eliminado posteriormente).

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ touch READMI.md

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ touch LICENCE.txt

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ touch passwords.txt

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ |
```

	LICENCE.txt	08/02/2025 9:35	Documento de tex...	0 KB
	passwords.txt	08/02/2025 9:35	Documento de tex...	0 KB
	READMI.md	08/02/2025 9:35	Markdown File	0 KB

Les añadimos contenido y para mostrar el estado del repositorio usamos el comando **git status** y observamos que al estar incluidos en el fichero **.gitignore** no aparecen.

```
echo "Este es el README del proyecto" >> README.md
echo "Esta es la licencia del proyecto" >> LICENCE.txt
echo "Este es un archivo de contraseñas" >> passwords.txt
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ echo "Este es el README del proyecto" >> README.md
$ echo "Esta es la licencia del proyecto" >> LICENCE.txt
$ echo "Este es un archivo de contraseñas" >> passwords.txt

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status
On branch main
nothing to commit, working tree clean

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

Para ver el estado de los archivos ignorados usamos el comando:

```
git status --ignored
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status --ignored
On branch main
Ignored files:
  (use "git add -f <file>..." to include in what will be committed)
    LICENCE.txt
    README.md
    passwords.txt

nothing to commit, working tree clean

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

Aunque el ejercicio no lo pide, se utilizará `git add`. para guardar los cambios en el área de preparación y se realizara un commit.

```
git add .
git commit -m "Añadido archivos README.md , LICENCE.txt y passwords.txt"
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git add .

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git commit -m "Añadido archivos README.md , LICENCE.txt y passwords.txt"
On branch main
nothing to commit, working tree clean
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status
On branch main
nothing to commit, working tree clean

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git log --oneline
1868a10 (HEAD -> main) Añadido .gitignore para README.md , LICENCE.txt y passwords.txt
34a367c Copiados archivos en labdist

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

4.Crea una rama **feature-estilos** . Cámbiate a ella.

-Modifica el archivo **estilos.css** :

- propiedad color del **body** y de **h2** : **#2a2a2a**
- propiedad **background-color** de **header** y **footer**:

#2a75ff

-Comprueba el estado del repositorio. Añade los cambios, realiza un commit con el mensaje "actualizados estilos a azules"

Para crear la rama usamos el comando `git branch`, usamos `git branch -av` para mostrar las ramas del repositorio.

```
git branch feature-estilos
git branch -av
```



```
MINGW64:/c:/Users/fuejo/OneDrive/Documentos/labdist
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git log --graph --oneline
* 1868a10 (HEAD -> main) Añadido .gitignore para README.md , LICENCE.txt y passwords.txt
* 34a367c Copiados archivos en labdist

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git branch feature-estilos

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git branch -av
feature-estilos 1868a10 Añadido .gitignore para README.md , LICENCE.txt y passwords.txt
* main          1868a10 Añadido .gitignore para README.md , LICENCE.txt y passwords.txt

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

Para cambiarme de rama usamos el comando:

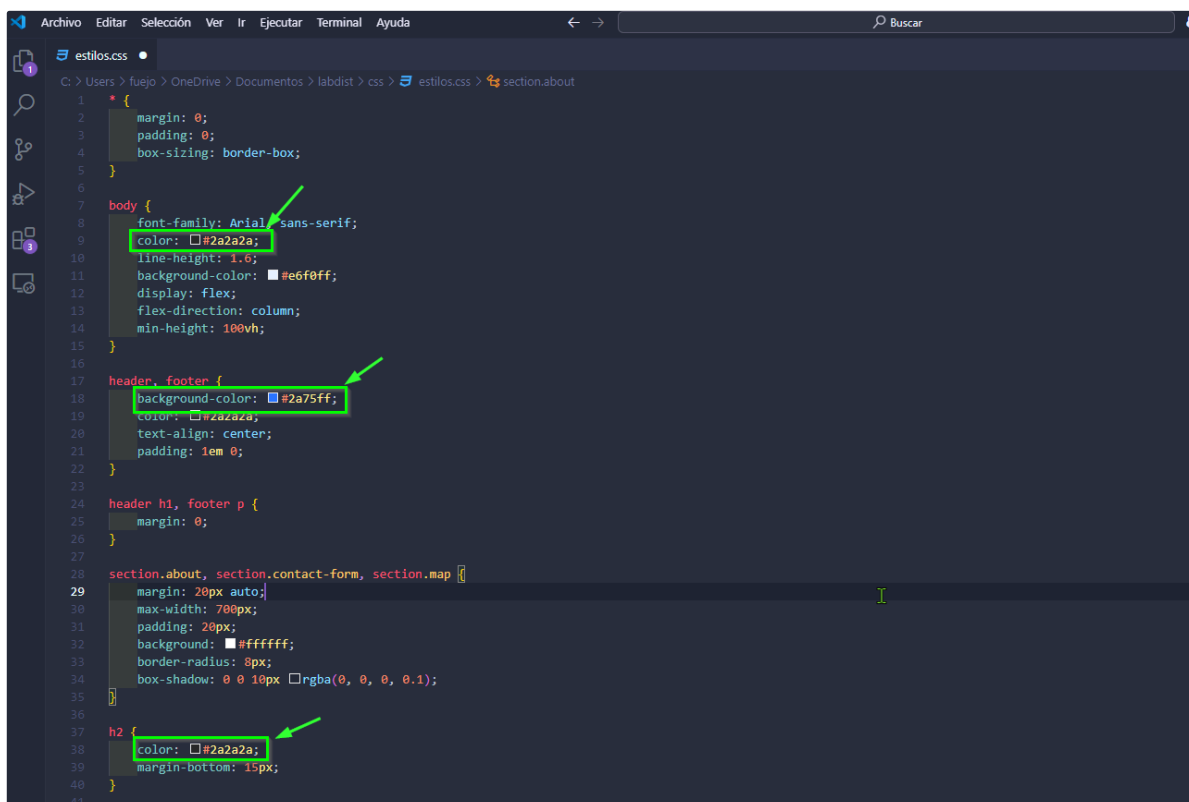
```
git checkout feature-estilos
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git branch -av
feature-estilos 1868a10 Añadido .gitignore para README.md , LICENCE.txt y passwords.txt
* main          1868a10 Añadido .gitignore para README.md , LICENCE.txt y passwords.txt

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git checkout feature-estilos
Switched to branch 'feature-estilos'

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (feature-estilos)
$
```

Para modificar la hoja de estilos.css se ha utilizado el IDE Visual Estudio:



Una vez modificada la hoja de estilos comprobamos el estado de estado del repositorio, añadimos los cambios al área de preparación y realizamos el commit.

```
git status
git add .
git commit -m "actualizados estilos a azules"
```

```
MINGW64: c:/Users/fuejo/OneDrive/Documentos/labdist

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (feature-estilos)
$ git status
On branch feature-estilos
nothing to commit, working tree clean

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (feature-estilos)
$ git status
On branch feature-estilos
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   css/estilos.css

no changes added to commit (use "git add" and/or "git commit -a")

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (feature-estilos)
$ git add .

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (feature-estilos)
$ git commit -m "actualizados estilos a azules"
[feature-estilos 4c3d6ba] actualizados estilos a azules
1 file changed, 3 insertions(+), 3 deletions(-)

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (feature-estilos)
$ git status
On branch feature-estilos
nothing to commit, working tree clean

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (feature-estilos)
$
```

5. Vuelve a la rama **main**. En el archivo **index.html** añade un comentario donde se indique tu nombre como autor de la página. Comprueba el estado del repositorio. Añade los cambios, realiza un commit con el mensaje 'añadido autor en index'. Muestra los logs del repositorio en una línea, gráficamente y con 'decoración'

Para volver a la rama **main** usamos:

```
git checkout main
```

Luego con Visual Estudio Code se añade el comentario indicando el autor:



```
Archivo Editar Selección Ver Ir Ejecutar ... Buscar
index.html 1 • Notas de la versión: 1.97.0
C:\Users\>fuejo>OneDrive>Documentos>labdist>index.html > html > body > section.about > ul > li
1 <!-- Autor: Adrián García de La Cera -->
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Labowebs - Sitios Web para Pequeñas Empresas</title>
8   <link rel="stylesheet" href="css/estilos.css">
9 </head>
10 <body>
11   <header>
12     <h1>Labowebs</h1>
13     <p>Confección de sitios web modernos y accesibles para pequeñas
    empresas</p>
```

Una vez guardado la modificación en Visual Estudio Code, comprobamos el estado del repositorio con **git status**, añadimos los cambios al área de preparación y realizamos el commit con el nombre indicado.

```
git status
git add .
git commit -m "añadido autor en index"
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git add .

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git commit -m "añadido autor en index"
[main e8a3bad] añadido autor en index
1 file changed, 1 insertion(+)

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git status
On branch main
nothing to commit, working tree clean

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

Procedemos a mostrar los log en línea, gráficamente y decoración:

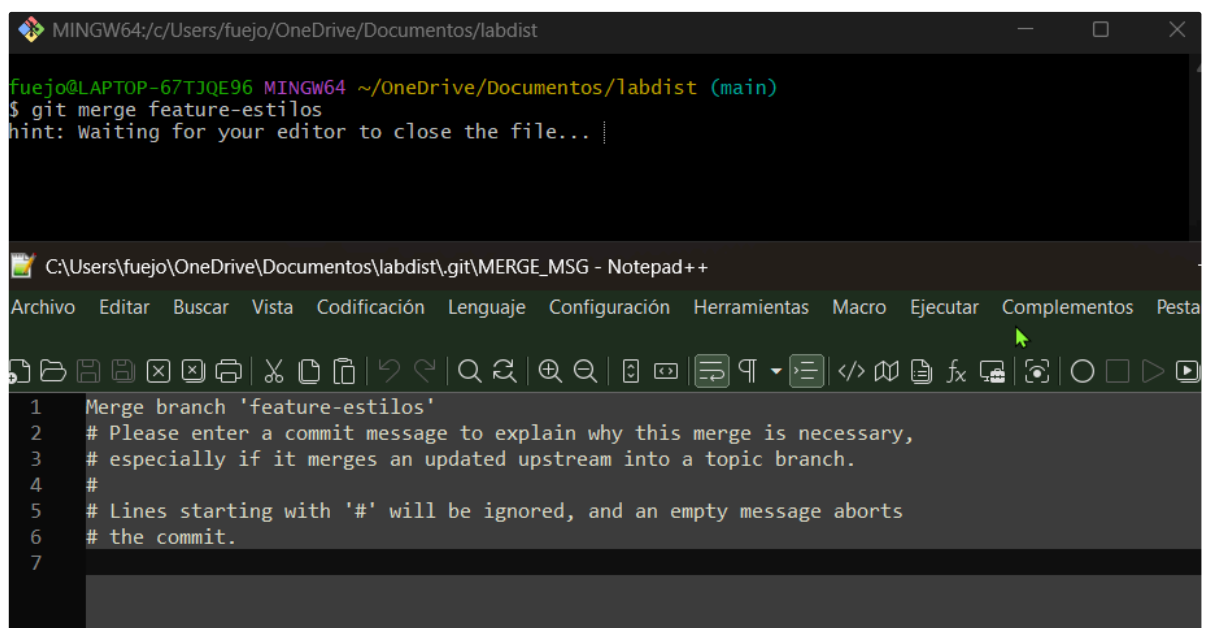
```
git log --graph --oneline --all --decorate
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git log --graph --oneline --all --decorate
* e8a3bad (HEAD -> main) añadido autor en index
| * 4c3d6ba (feature-estilos) actualizados estilos a azules
|/
* 1868a10 Añadido .gitignore para README.md , LICENCE.txt y passwords.txt
* 34a367c Copiados archivos en labdist
```

6.Fusiona la rama **feature-estilos** en la rama **main** . Muestra los logs del repositorio en una línea, gráficamente y con 'decoración'

Para fusionar la rama **feature-estilos** y **main** se utiliza el comando **git merge**, al realizar la fusión de ramas se realiza automáticamente un commit, como al realizar commit se le debe dar un nombre, al lanzar el comando nos abrirá el editor de texto y nos indicará un nombre por defecto, el cual podemos dejar o modificar.

```
git merge feature-estilos
```



Una vez realizado mostramos los log con el comando:

```
git log --graph --oneline --all --decorate
```

```
MINGW64:/c/Users/fuejo/OneDrive/Documentos/labdist

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git merge feature-estilos
Merge made by the 'ort' strategy.
 css/estilos.css | 6 +++---
 1 file changed, 3 insertions(+), 3 deletions(-)

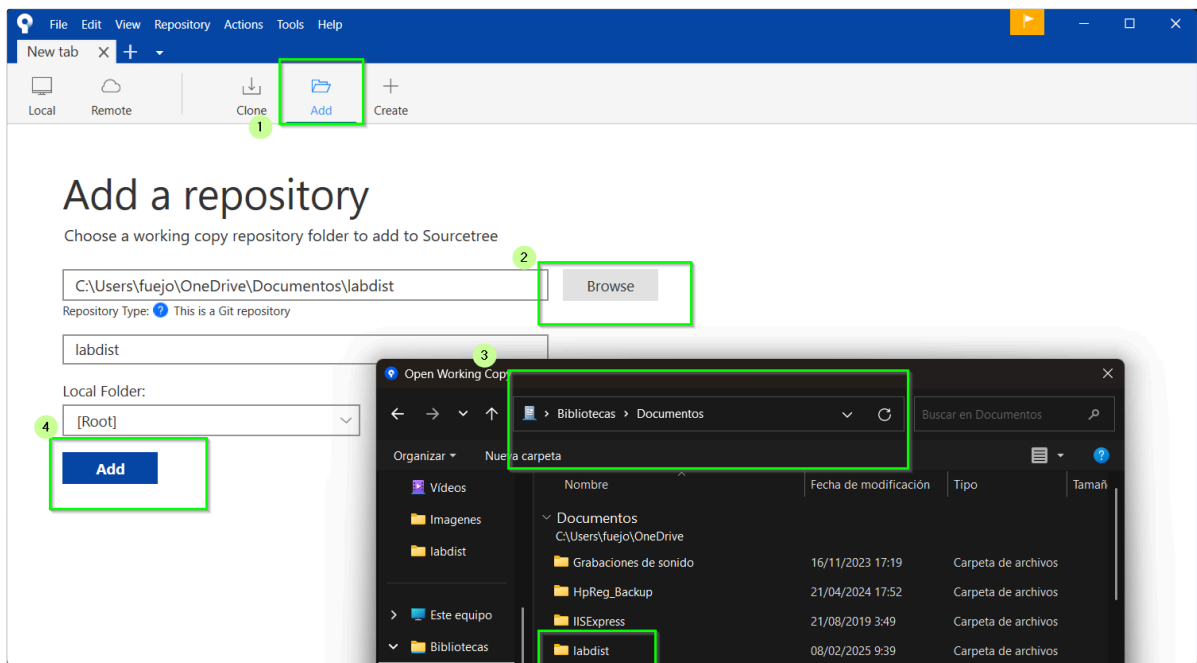
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git log --graph --oneline --all --decorate
* 4f30d38 (HEAD -> main) Merge branch 'feature-estilos'
|
| * 4c3d6ba (feature-estilos) actualizados estilos a azules
| * e8a3bad añadido autor en index
|/
* 1868a10 Añadido .gitignore para README.md , LICENCE.txt y passwords.txt
* 34a367c Copiados archivos en labdist

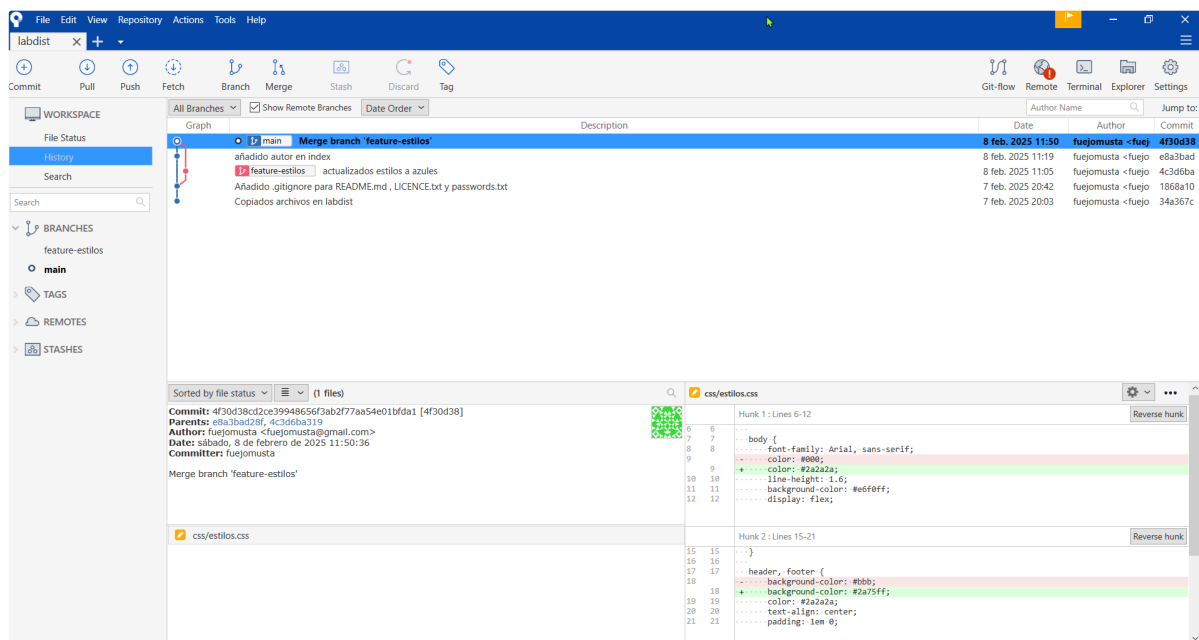
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ |
```

Trabajo en remoto

1.Continúa con el repositorio **labdist** . Añade el repositorio a Sourcetree.

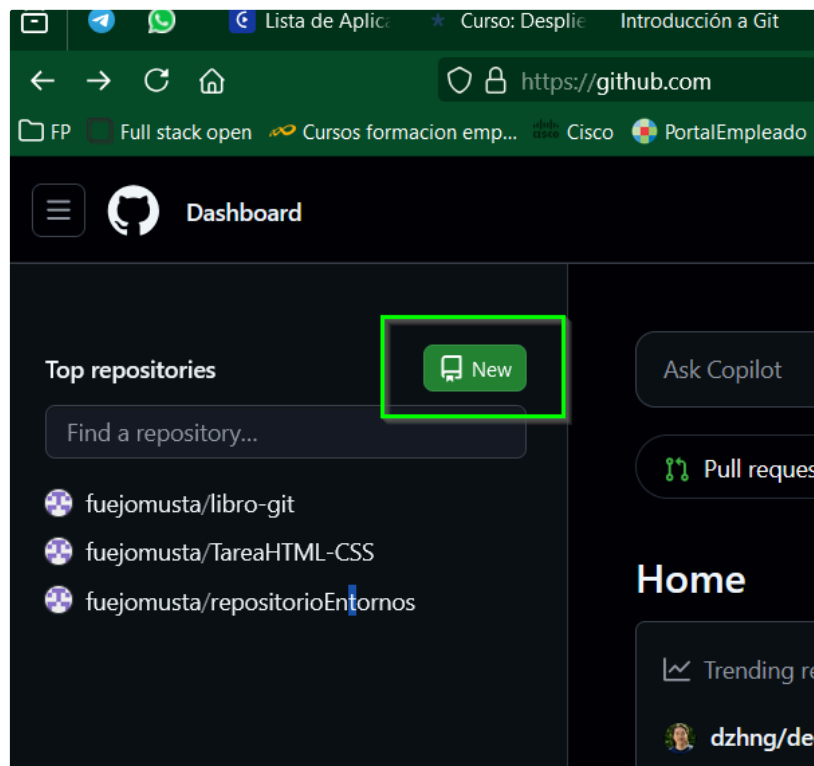
Para añadir el repositorio **labdist** a Sourcetree una vez abierto el programa, seleccionamos add, despues en browse buscamos nuestro repositorio, lo seleccionamos y pulsamos el botón Add.





2. En tu cuenta de GitHub, crea un repositorio remoto y sube al remoto los ficheros de tu repositorio local. Debes subir todas las ramas. Muestra, además, la captura de pantalla donde se vean en GitHub.

Para crear el repositorio, accedemos a nuestra cuenta GitHub, y seleccionamos el botón New que se encuentra en la parte izquierda de la pantalla.




Una vez dado a new, debemos completar unos campos como su nombre (labdistAdriánTarea2), añadimos una pequeña descripción, nos aseguramos sea publico para la tarea.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *


 fuejomusta / labdistAdriánTarea2


Your new repository will be created as labdistAdri-nTarea2.
The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. Need inspiration? How about [cautious-garbanzo](#) ?

Description (optional)

Tarea Desarrollo aplicaciones web labdist :Ejercicio de Git - proyecto labdist

☒  Public
Anyone on the internet can see this repository. You choose who can commit.

☐  Private
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

Una vez creado el repositorio procedemos a subir las ramas del repositorio local a GitHub, para ello debemos enlazar el repositorio local con el remoto con `git remote add origin`, con `git remote -v` verificamos que se ha agregado correctamente y para subir todas las ramas se usa `git push -u origin --all`, en mi caso me ha dado error pues la autenticación no era la recomendada por lo que he creado un token en GitHub para poder dar acceso

```
git remote add origin https://github.com/fuejomusta/labdistAdri-nTarea2.git
git remote -v
git push -u origin --all
```

```
MINGW64:/c:/Users/fuejo/OneDrive/Documentos/labdist
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git remote add origin https://github.com/fuejomusta/labdistAdri-nTarea2.git

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git remote -v
origin https://github.com/fuejomusta/labdistAdri-nTarea2.git (fetch)
origin https://github.com/fuejomusta/labdistAdri-nTarea2.git (push)

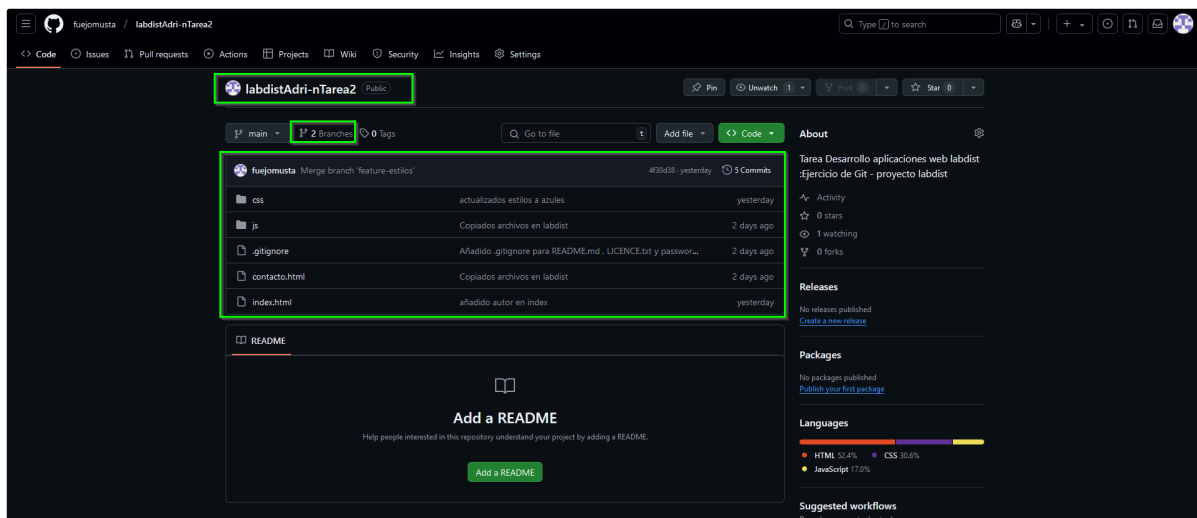
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git push -u origin --all
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-reposito
ries#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/fuejomusta/labdistAdri-nTarea2.git/'
```

```
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git push -u origin --all
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 3.46 KiB | 590.00 KiB/s, done.
Total 20 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7/7), done.
To https://github.com/fuejomusta/labdistAdri-nTarea2.git
 * [new branch]      feature-estilos -> feature-estilos
 * [new branch]      main -> main
branch 'feature-estilos' set up to track 'origin/feature-estilos'.
branch 'main' set up to track 'origin/main'.

fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$ git remote -v
origin https://github.com/fuejomusta/labdistAdri-nTarea2.git (fetch)
origin https://github.com/fuejomusta/labdistAdri-nTarea2.git (push)

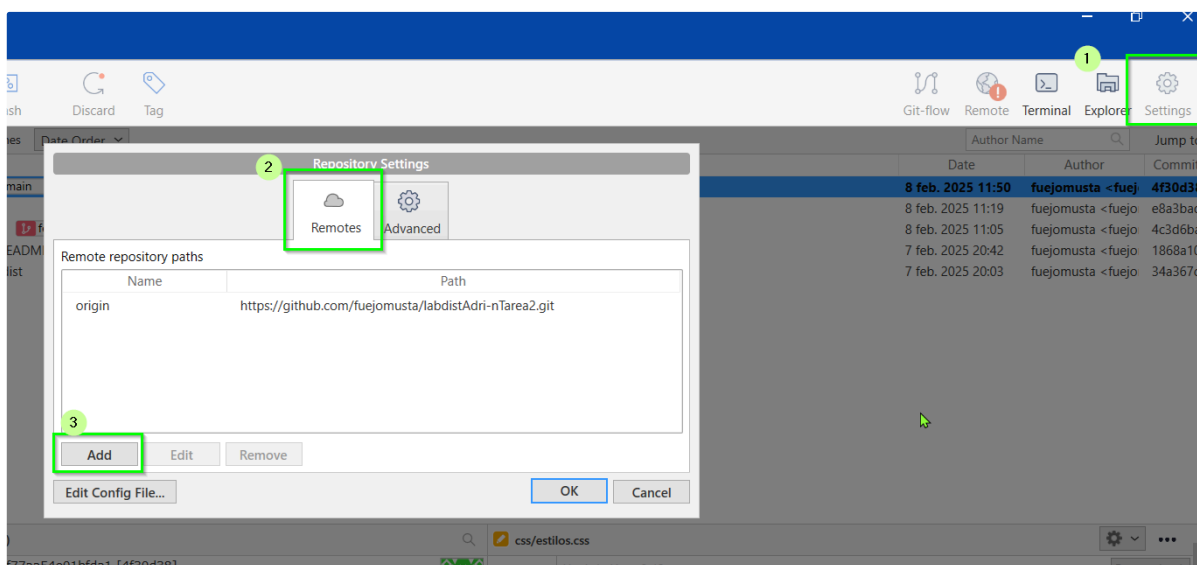
fuejo@LAPTOP-67TJQE96 MINGW64 ~/OneDrive/Documentos/labdist (main)
$
```

Una vez realizado, comprobamos en la web que se han subido los archivos del repositorio y sus ramas.



** Esta parte se ha realizado en consola de Git Bash por error, para realizarlo desde SourceTree habría que seguir los siguientes pasos:

1. Seleccionar Settings
2. En pestaña remotos
3. Pulsar botón add
4. En nombre ponerle el nombre del repositorio y en URL añadir el link de enlace de GitHub



Remote details


4 Required information

Remote name: ☐ Default remote

URL / Path:

Optional extended integration

Remote Account:

 **Generic Account**
Generic Host

Legacy Account Settings:

Host Type:






Host Root URL:

Username:

Extended integration is used to enable deeper integration with hosting providers such as Bitbucket, including locating existing clones when following links from sites and creating pull requests.

3. En el repositorio local, crea una rama **feature-index**. Añade el siguiente código dentro de la **<section class="about">**. Añade los cambios y crea un commit con el mensaje "Añadido párrafo equipo en index.html". Sube los cambios al remoto. (Recuerda que debes usar SourceTree en todo este apartado)



Para crear la rama desde Sourcetree, hacemos clic en el botón branch, nos abrirá una ventana en la que indicamos el nombre de la rama y hacemos clic en el botón Create Branch

Branch Merge Stash Discard Tag

☒ Show Remote Branches

Branch

 New Branch  Delete Branches

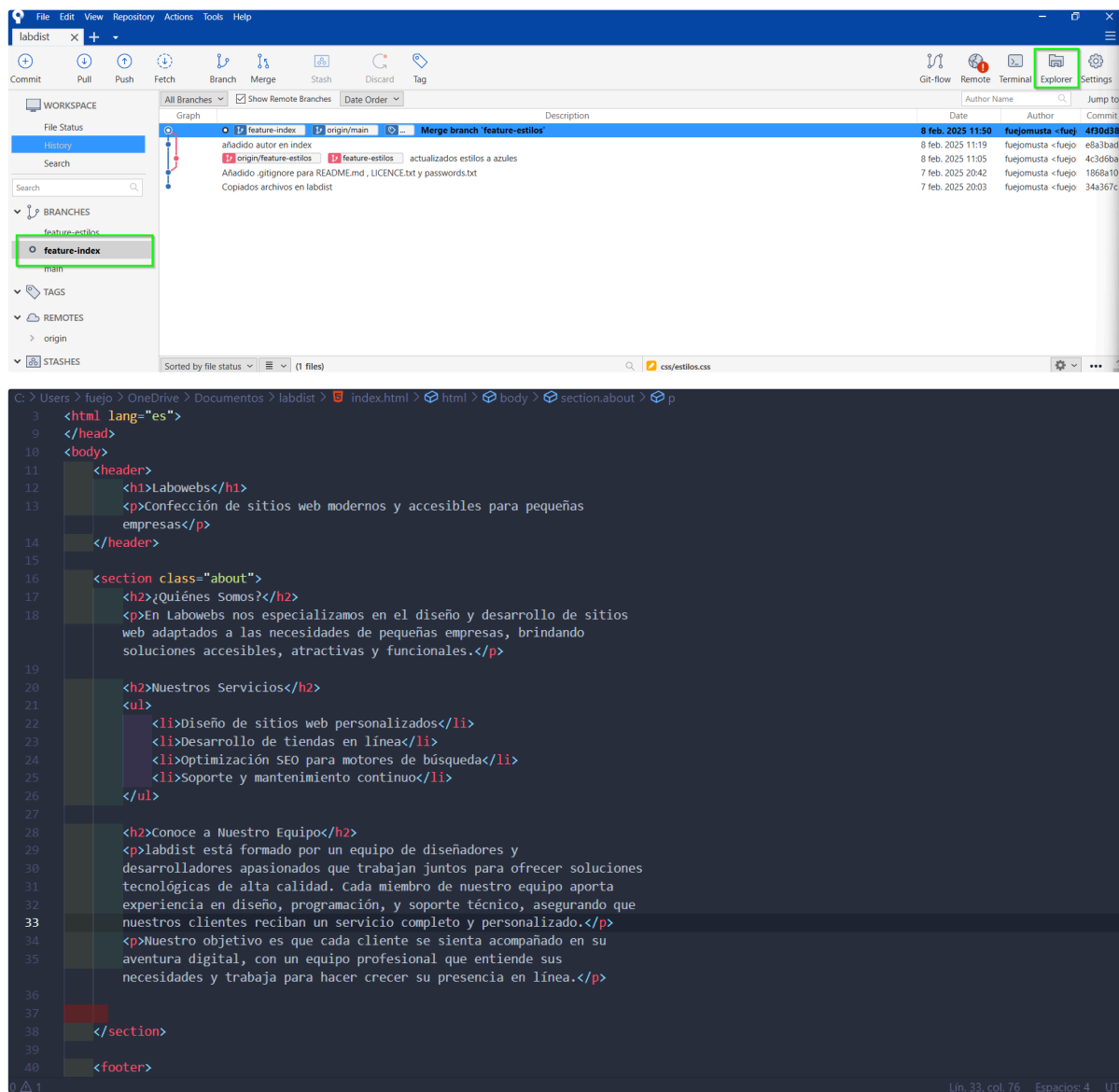
Current Branch: main

New Branch:

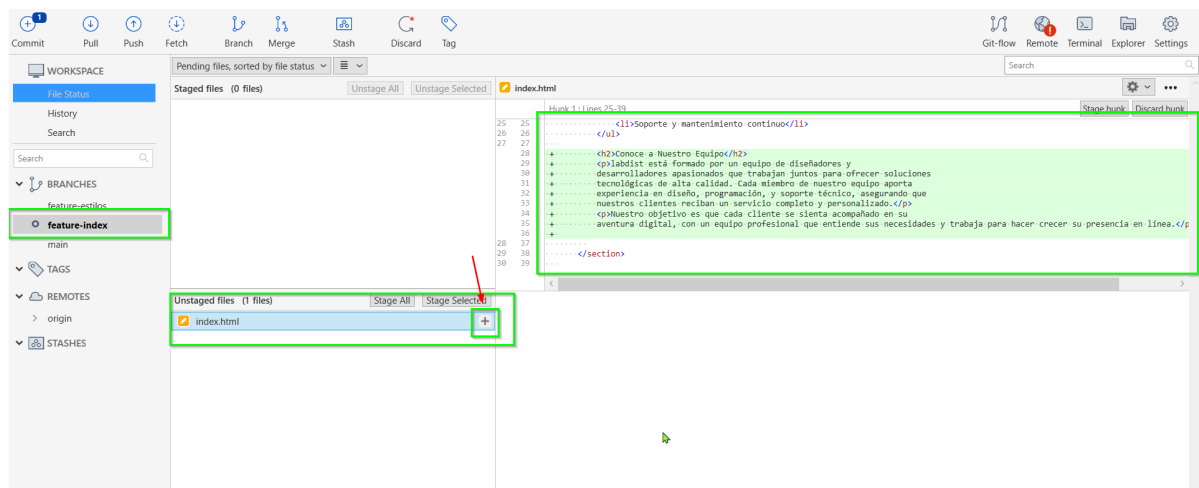
Commit: ☒ Working copy parent
☐ Specified commit:

☒ Checkout New Branch

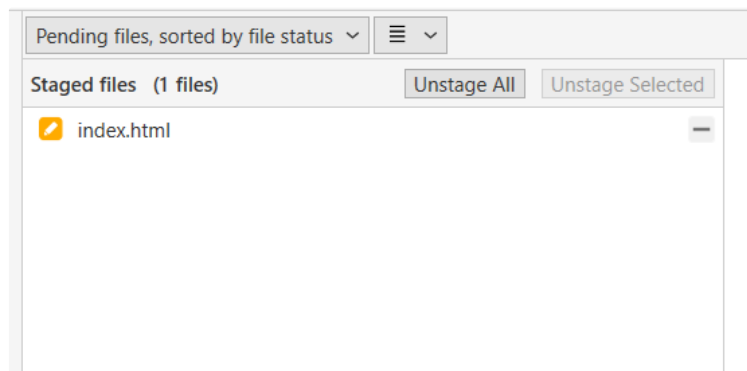
Una vez creada la rama, asegurándonos de estar en dicha rama, hacemos clic en el botón explorer de Sourcetree, nos abrirá un explorador de archivos en los que vemos solo los archivos desde los que podemos acceder desde dicha rama, y modificamos el archivo **index.html** como se indica en el enunciado.



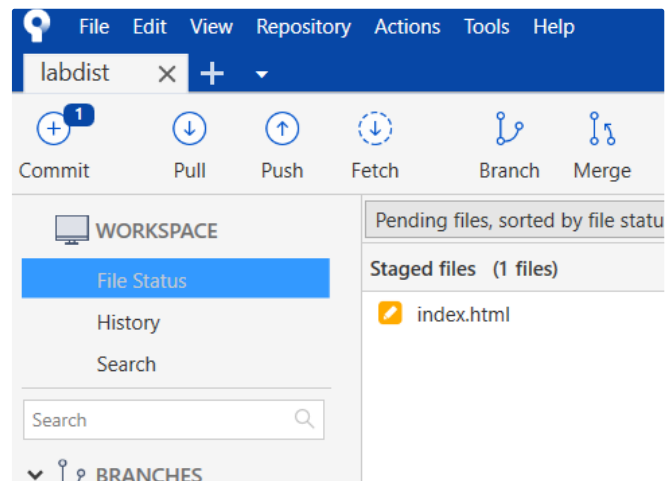
Una vez realizados los cambios, en File Status nos muestra que hay cambios en el archivo index.html en la ventana **Unstaged files**, si lo seleccionamos nos muestra los cambios realizados en el archivo. Los añadimos a través del botón + que tiene a su derecha.



Una vez dado al más el archivo queda en el area de preparación, en Sourcetree te lo muestra en Staged files.



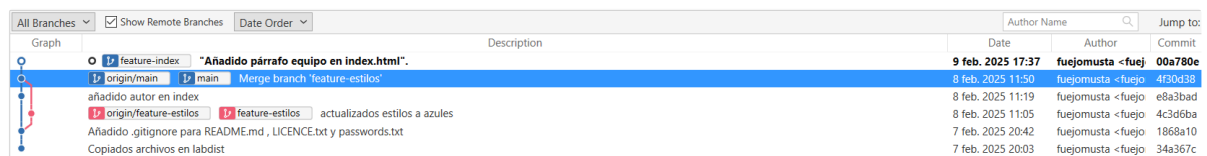
Ahora para hacer el commit, simplemente pulsamos el botón de Commit, el cual además nos indica que tenemos cambios que commitear.



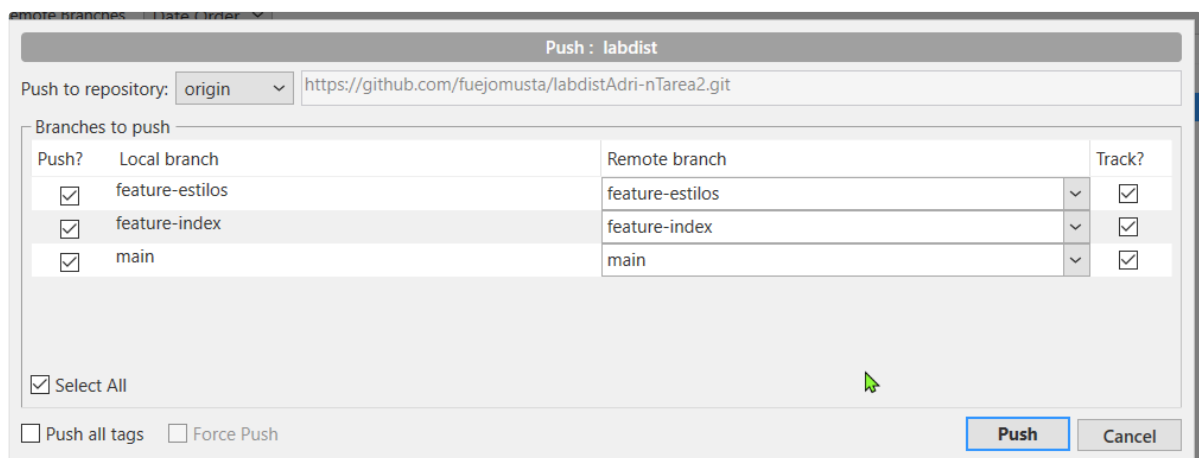
Añadimos el comentario y hacemos el commit.



Una vez realizado el commit, nos muestra el resultado.



Ahora es necesario subir los cambios al repositorio remoto con el botón push.

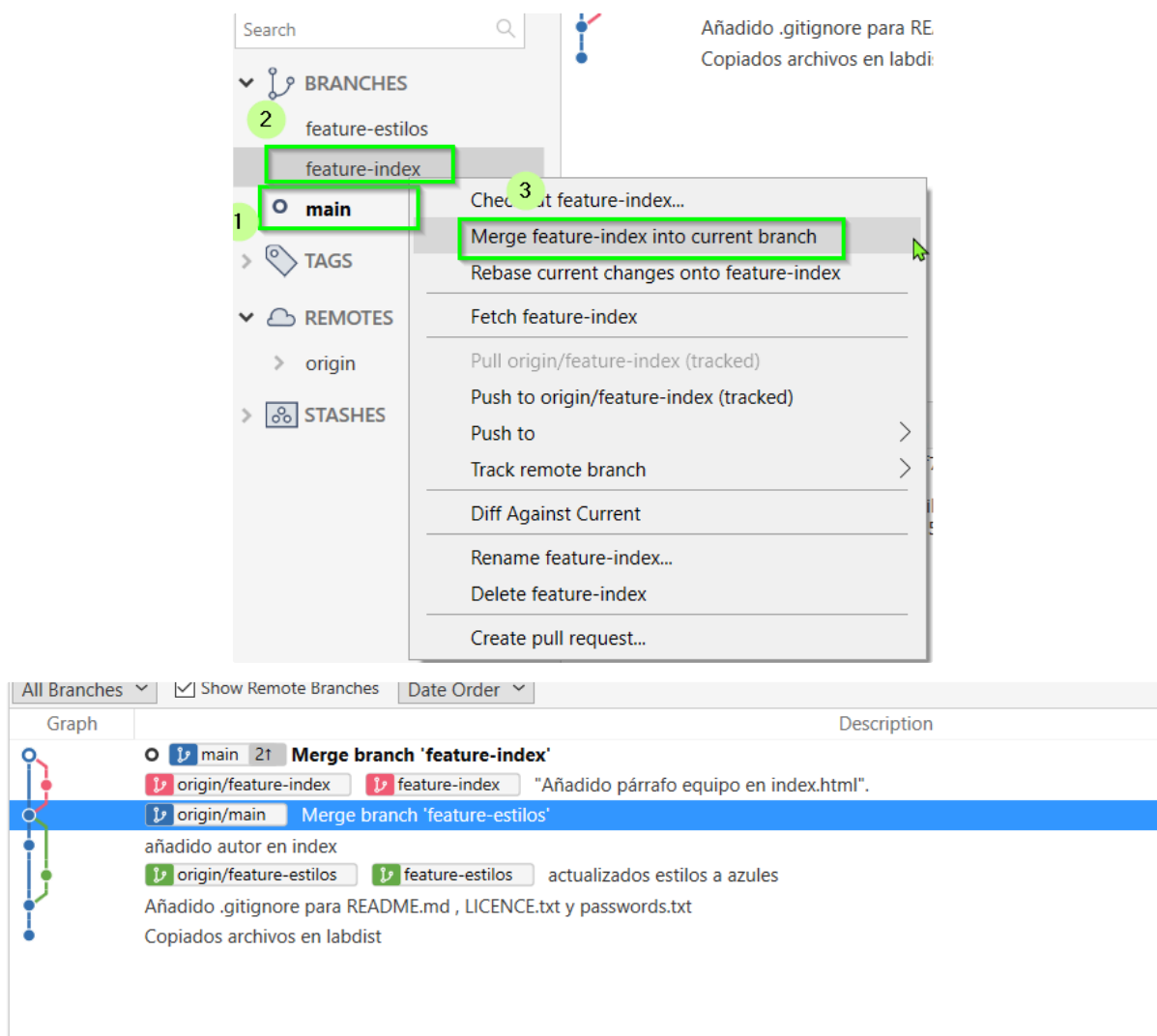


Comprobamos en GitHub que ha subido la rama



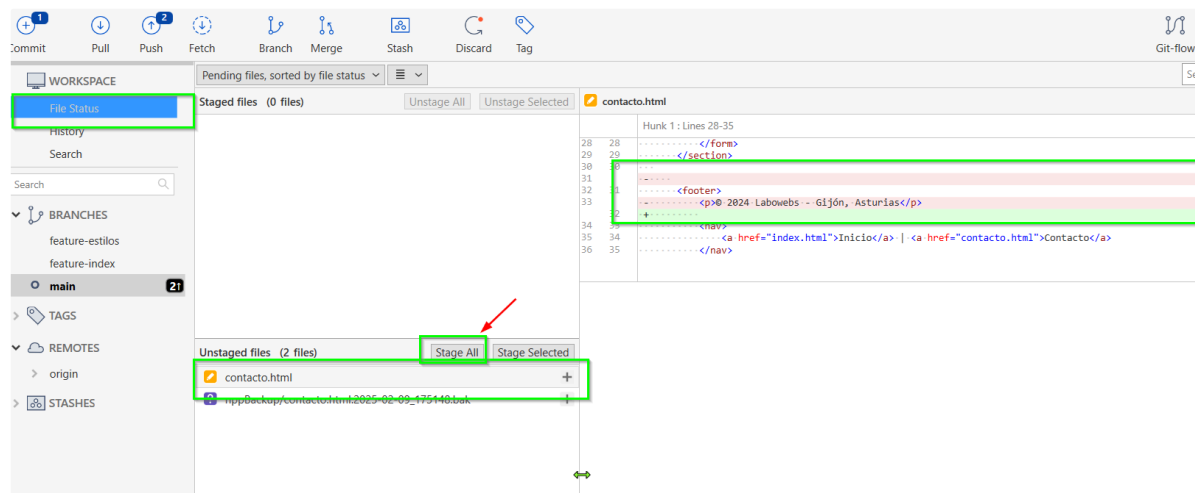
4. En el repositorio local, fusiona la rama **feature-index** en la rama **main**.

Para fusionar las ramas desde sourcetree seleccionamos la rama principal **main**, hacemos clic con botón derecho sobre la rama a fusionar **feature-index** y seleccionamos Merge feature-index.

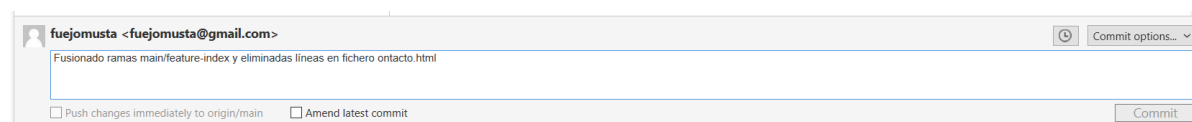


5. Editar el fichero **contacto.html**. Borrar unas líneas. Muestra los ficheros con cambios pendientes y las diferencias. Añade los cambios y haz un commit.

Tras acceder al fichero **contacto.html** y borrar unas líneas comprobamos el estado del archivo en Sourcetree, al seleccionarlo nos indica las líneas modificadas, usamos el botón Stage All para añadir los cambios al área de preparación.

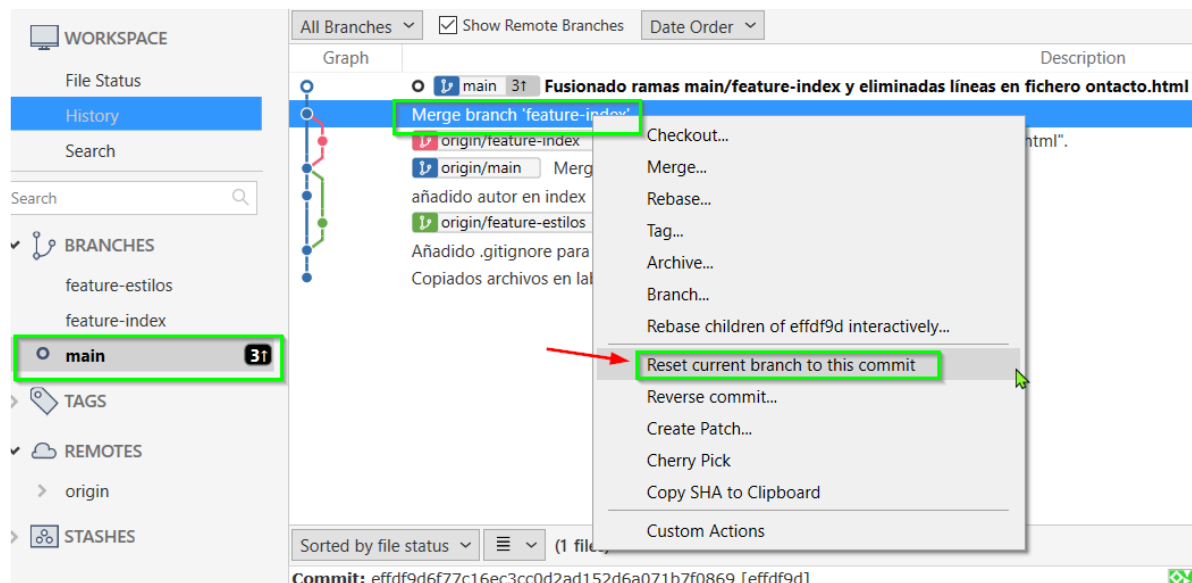


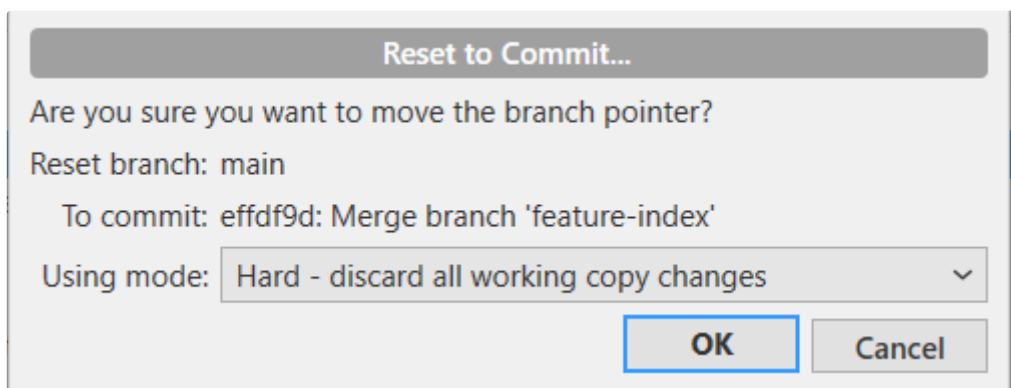
Y realizamos el commit.



6. Te das cuenta del error. Deshaz TOTALMENTE el commit anterior. Captura el estado actual del repositorio. (Asegúrate de que el fichero **contacto.html** ha recuperado todas las líneas borradas y no hay cambios pendientes en el repositorio.)

Para deshacer los cambios, me aseguro de tener seleccionada la rama **main**, selecciono el anterior cambio que es en el que se fusionaron la rama **main** y **feature-index** con botón derecho y selecciono Reset current branch y selecciono Hard reset





Comprobamos que el fichero ha recuperado las líneas que se habían eliminado.

```

26
27     <button type="submit">Enviar</button>
28 </form>
29 </section>
30
31
32 <footer>
33     <p>© 2024 Labowebs - Gijón, Asturias</p>
34     <nav>
35         <a href="index.html">Inicio</a> | <a href="contacto.html">Contacto</a>
36     </nav>
37 </footer>
38 </body>
39 </html>
40

```

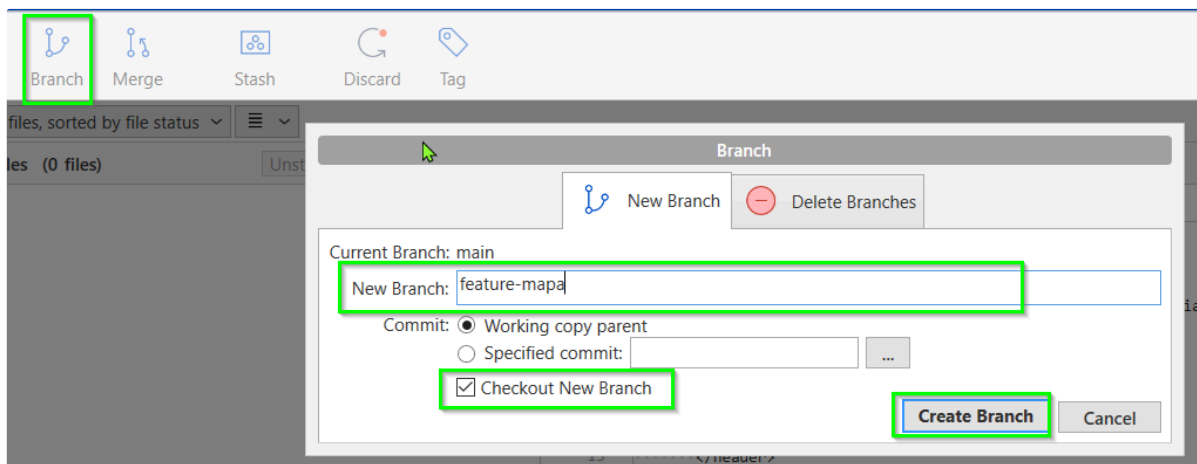
8. Crea una rama **feature-mapa** y cámbiate a ella. Incluye este código en el archivo **contacto.html**. Añade los cambios. Realiza un commit.

```

<section class="map">
<h2>Nuestra Ubicación</h2>
<p>C/Luis Moya 335, Gijón, Asturias</p>
<iframe src="https://www.google.com/maps?
q=C%2FLuis%20Moya%20335%2C%20Gij%C3%B3n%2C%20Asturias&output=embed"
width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"></iframe>
</section>

```

Para crear la rama desde SourceTree, selecciono el botón Branch, indico el nombre indicado **feature-mapa**, me aseguro de tener seleccionado Checkout New Branch y selecciona create branch.



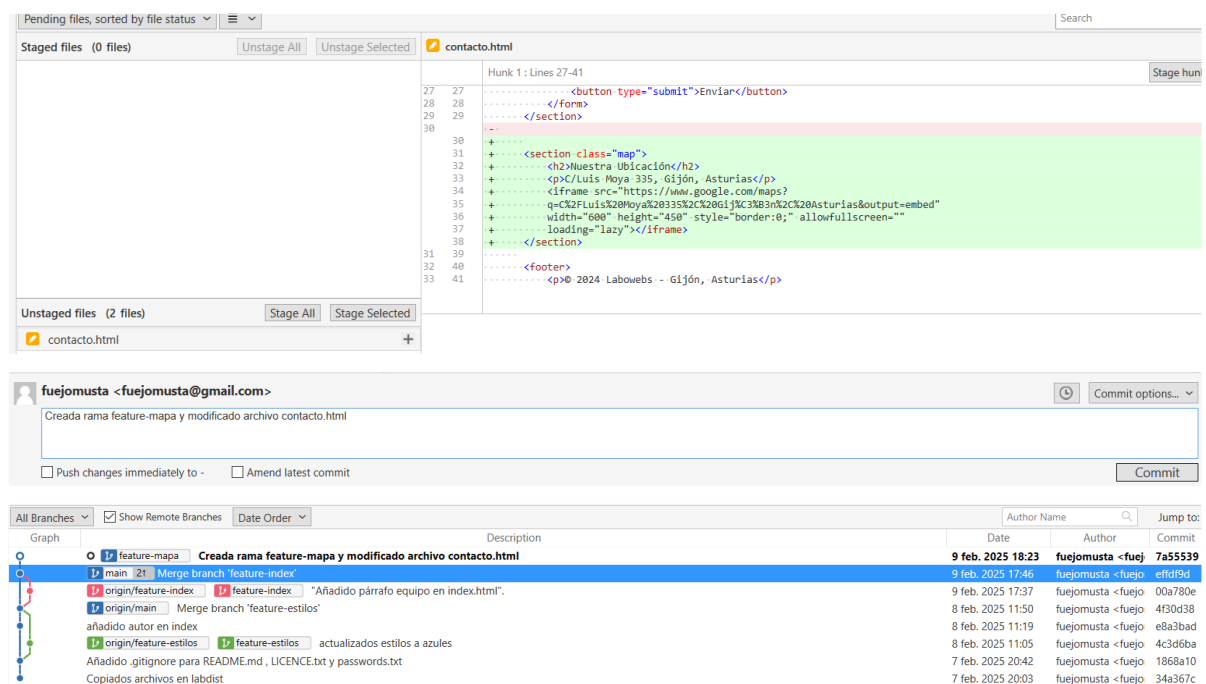
Una vez creada, me aseguro de tener seleccionada la nueva rama, accedo a contactos desde el explorador de Sourcetree y añado el código indicado antes del footer.

```
<button type="submit">Enviar</button>
</form>
</section>

<section class="map">
  <h2>Nuestra Ubicación</h2>
  <p>C/Luis Moya 335, Gijón, Asturias</p>
  <iframe src="https://www.google.com/maps?
q=C%2FLuis%20Moya%20335%2C%20Gij%C3%B3n%2C%20Asturias&output=embed"
width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"></iframe>
</section>

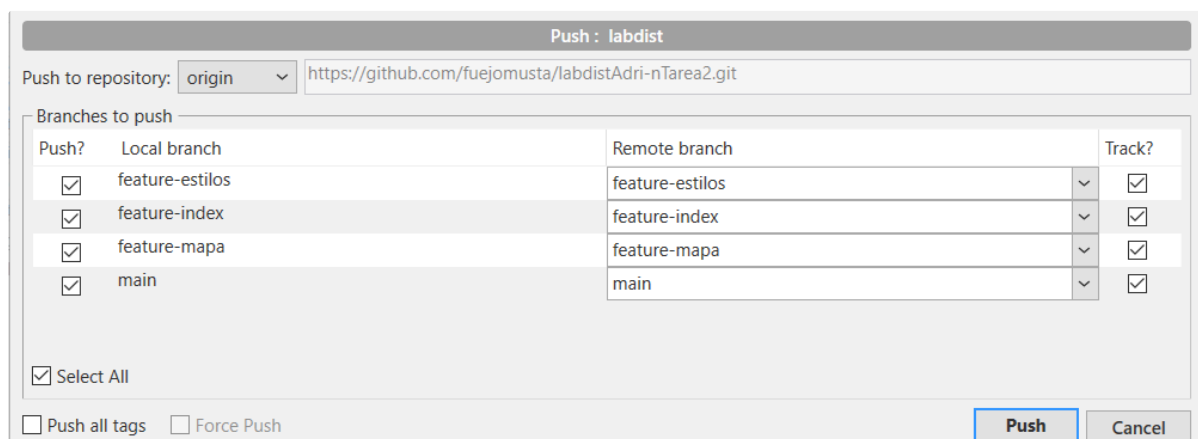
<footer>
  <p>© 2024 Labowebs - Gijón, Asturias</p>
  <nav>
```

Añado los cambios al area de preparación y realizo el commit

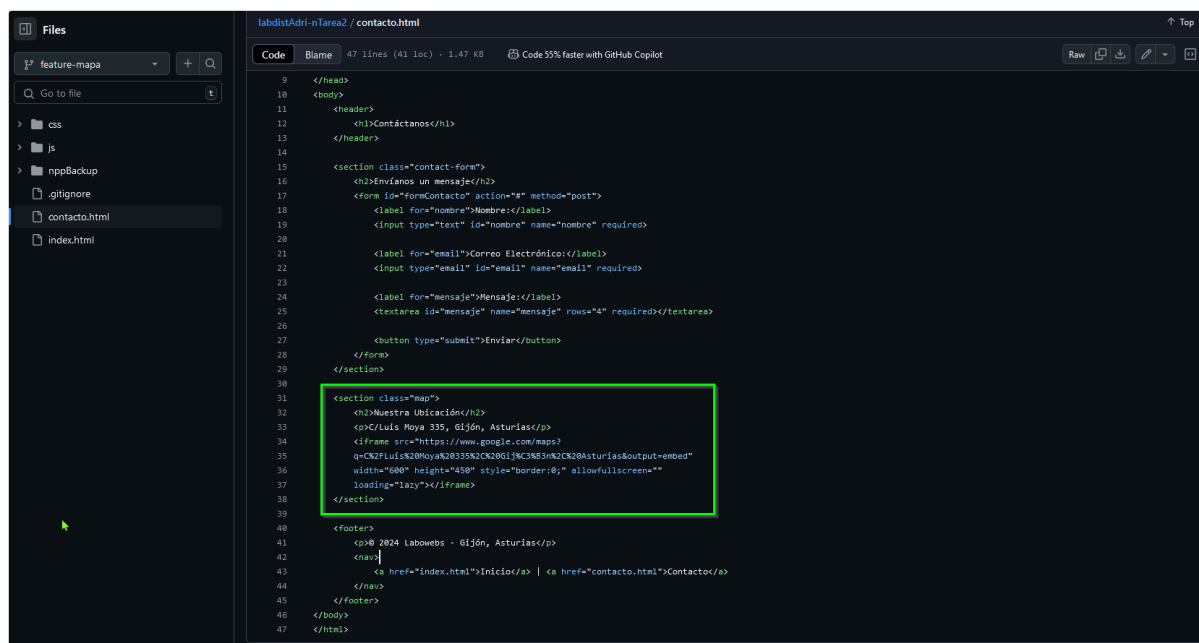
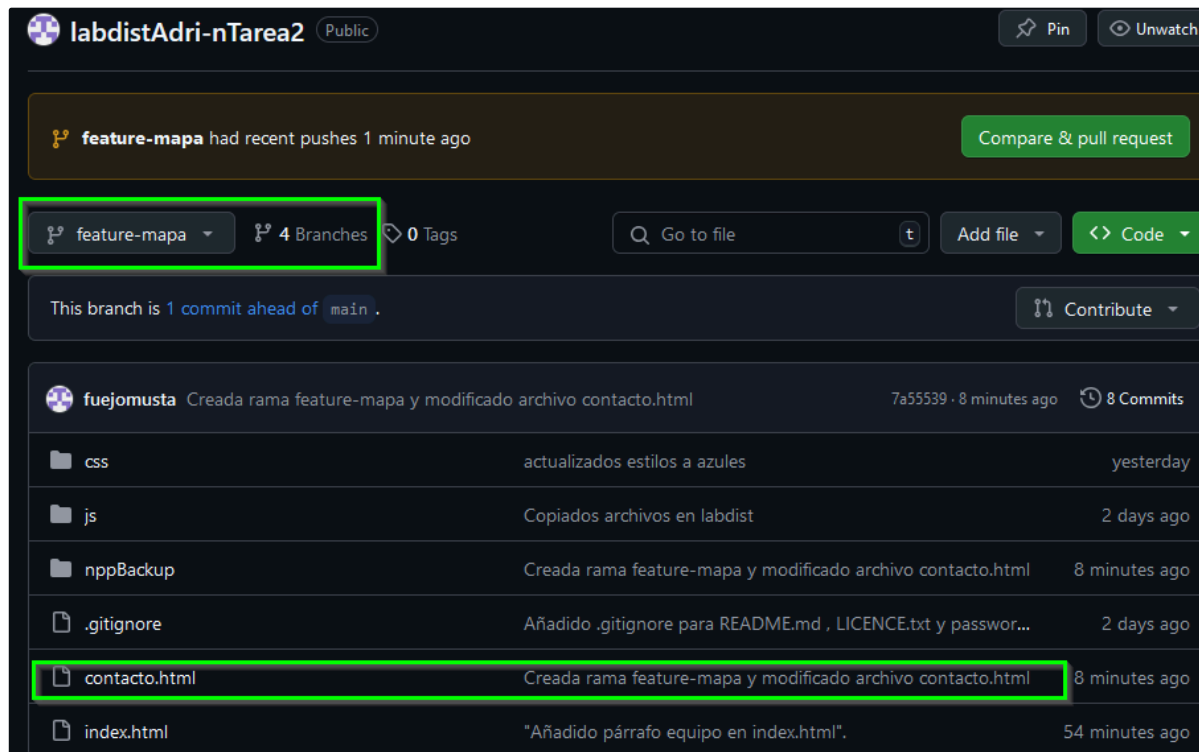


8. Sube los cambios al remoto - los de todas las ramas. Muestra en el remoto los cambios del archivo **contacto.html** en la rama **feature-mapa**.

Para subir los cambios al remoto uso el botón Push, selecciono todas las ramas y pulso Push.

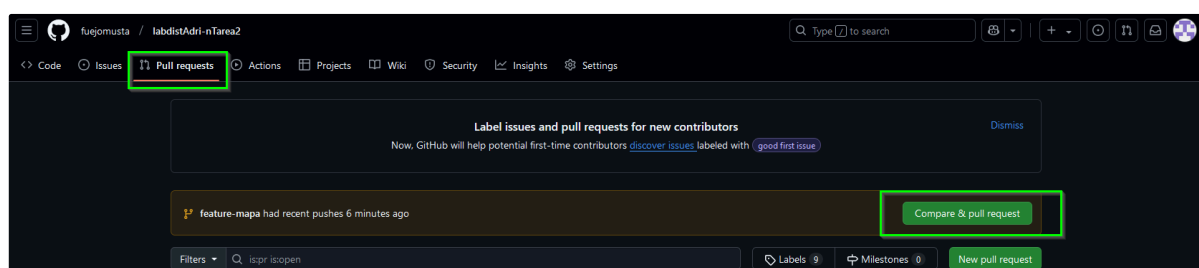


Compruebo en GitHub que se ha subido la nueva rama.



9. En GitHub, en la rama **main**, fusiona la rama **feature-mapa**. Baja los cambios del remoto a local. Deja los dos repositorios sincronizados. Muestra una captura de pantalla donde se vea la página principal de tu repositorio remoto

Para realizar la fusión, simularemos una revisión del código con Pull Request



Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff con](#)

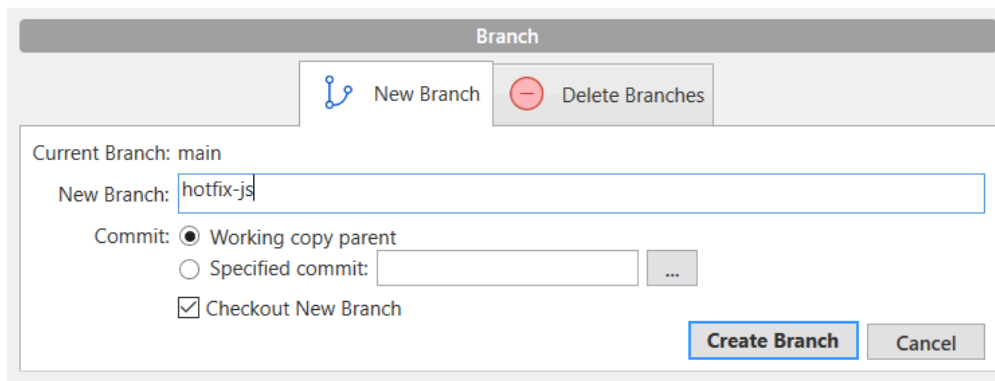
Tras indicar que cree el Pull request, automáticamente me lleva a la ventana en la que me indica que tengo 1 pull request pendiente y selecciono Merge pull request y confirmo el merge

Página 24 de 28

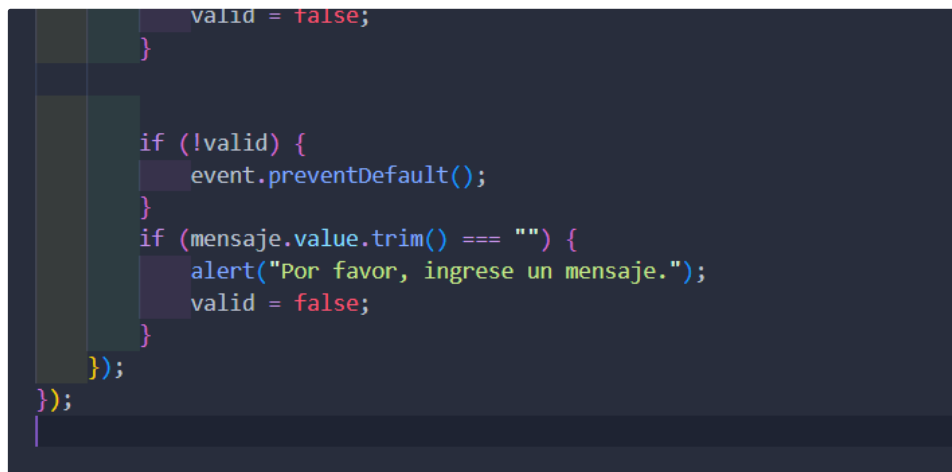
Conflictos

1.Crea una rama **hotfix-js** . Cámbiate a ella. Añade este código en el fichero **script.js** . Confirma el cambio y haz un commit con el mensaje "corregido problema en script.js".

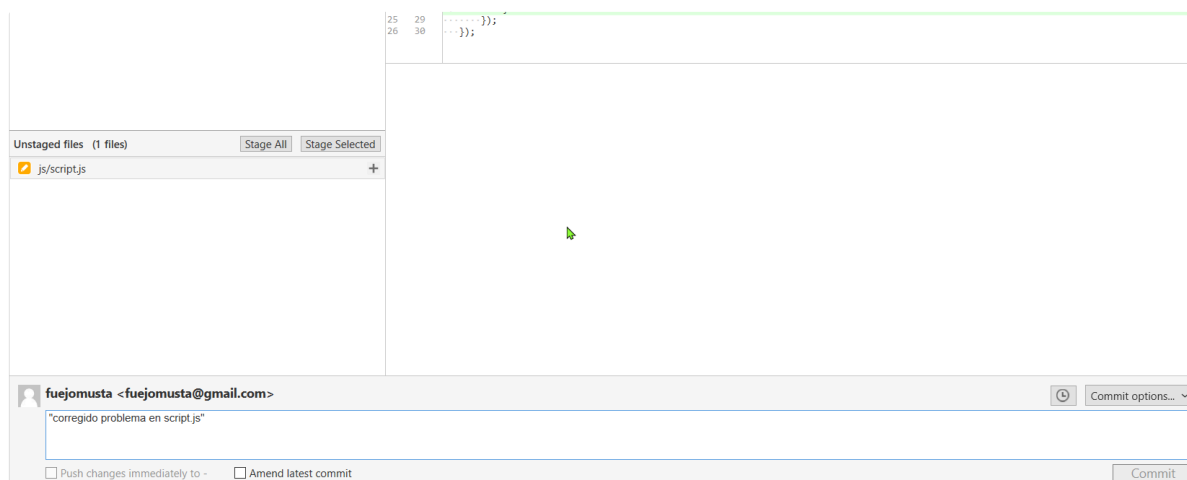
Para crear la rama seleccionamos el boton Branch.



Agregamos el código con visual estudio.

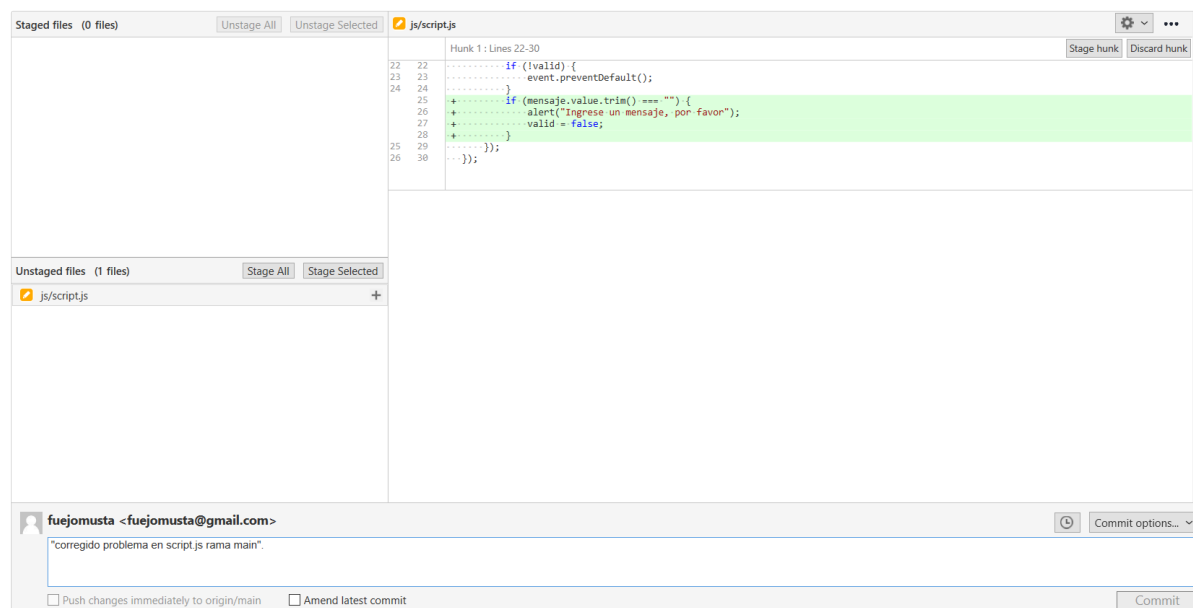


Subimos los cambios y realizamos el commit.



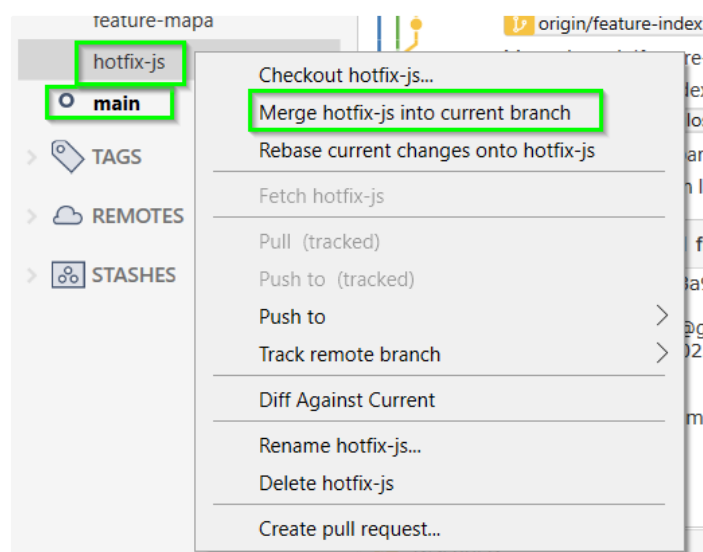
2. Vuelve a la rama **main** . En el fichero **script.js** en las mismas líneas que en la cuestión anterior, añade el código siguiente. Confirma el cambio y haz un commit con el mensaje "corregido problema en script.js rama main".

Realizo los mismos pasos indicados en el ejercicio anterior modificando el archivo y vuelvo a hacer commit.

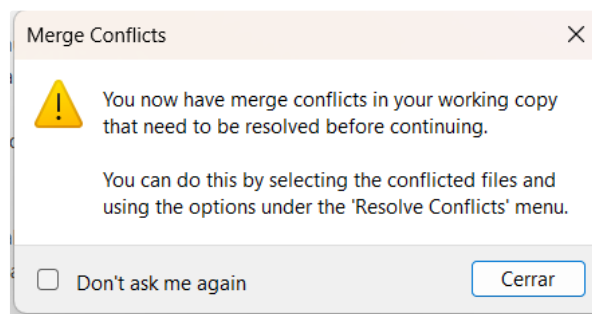


3. Fusiona la rama **hotfix-js** en **main** . Debe producirse un conflicto. Resuélvelo como consideres oportuno. Cuando termines la resolución del conflicto sube los cambios al remoto.

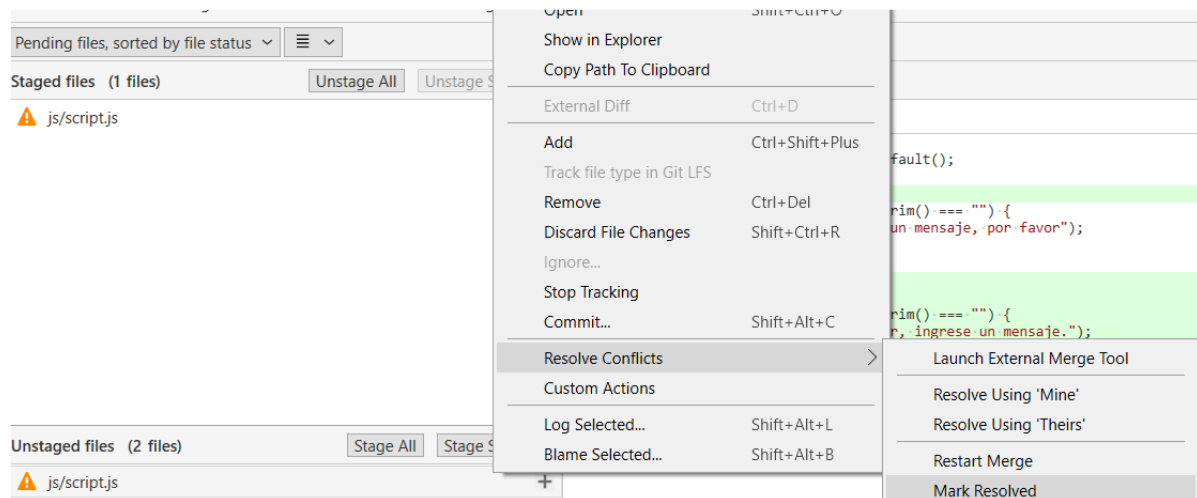
Para fusionar las ramas, selecciono la rama main, hago clic derecho sobre la rama hotfix-js y selecciono merge hotfix



Me da un error de conflicto:

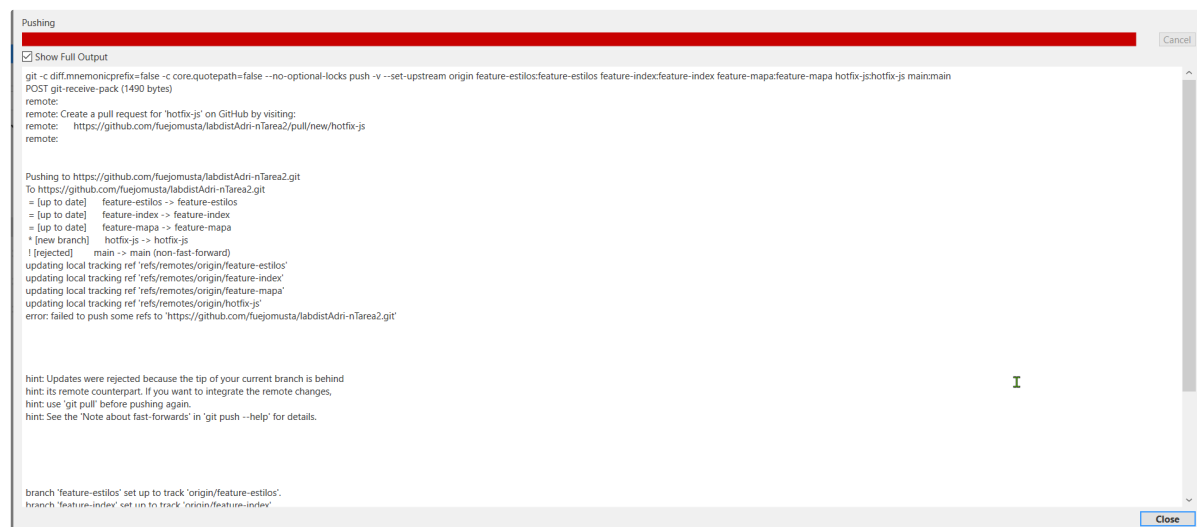


Para resolverlo abro de nuevo el archivo y elimino los cambios, guardo, vuelvo a Sourcetree y desde el main en estado de archivos, le indico con botón derecho que está resuelto.



Para finalizar realizo el commit y subo los cambios realizados.

Tras estos últimos cambios al intentar subir el repositorio remoto me ha dado un error



Tras realizar un Pull, revisar conflictos, he podido realizar el Push

