

Ejercicios de Git - labowebs

Ejercicios de Git - labowebs

Trabajo en local

Trabajo en remoto

Conflictos

Extra

Trabajo en local

1. Inicializa un nuevo repositorio Git en una carpeta llamada "labowebs" 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.

```
$ git init
$ git branch -m master main
$ git add .
$ git commit -m "Commit Inicial"
$ git log --oneline --all --graph --decorate
```

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs
$ git init
Initialized empty Git repository in C:/Users/alumno/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs/.git/

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (master)
$ git branch -m master main

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git add .

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git commit -m "Commit inicial"
[main (root-commit) 633903c] Commit inicial
4 files changed, 206 insertions(+)
create mode 100644 contacto.html
create mode 100644 css/estilos.css
create mode 100644 index.html
create mode 100644 js/script.js

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git log --oneline --all --graph --decorate
* 633903c (HEAD -> main) Commit inicial

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (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.

Usando visual studio code escribo lo siguiente dentro del `.gitignore` recién creado.

```
README.txt
LICENCE.txt
passwords.txt
```

```
$ git add .gitignore
$ git commit -m ""
$ git log --oneline --all --graph --decorate
```

```

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labow
ebs/laboweb (main)
$ git add .gitignore

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb (main)
$ git commit -m "Added README.txt, LICENCE.txt and passwords.txt to .gitignore"
[main 3a5fd55] Added README.txt, LICENCE.txt and passwords.txt to .gitignore
1 file changed, 3 insertions(+)
create mode 100644 .gitignore

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb (main)
$ git log --oneline --all --graph --decorate
* 3a5fd55 (HEAD -> main) Added README.txt, LICENCE.txt and passwords.txt to .gitignore
* 633903c Commit inicial

```

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.

```
$ git status
```

README.txt	14/11/2024 8:59	Documento de te...	0 KB
LICENCE.txt	14/11/2024 8:59	Documento de te...	0 KB
passwords.txt	14/11/2024 8:59	Documento de te...	0 KB

```

MINGW64/c:/Users/alumno/Documents/Ines/U4/ejercicioEntregable Git/laboweb/laboweb
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb/laboweb (main)
$ git status
On branch main
nothing to commit, working tree clean

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb/laboweb (main)
$

```

```
$ git ls-file --ignored --exclude-standard -o
```

```

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb/laboweb (main)
$ git ls-files --ignored --exclude-standard -o
LICENCE.txt
README.txt
passwords.txt

```

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

```
$ git checkout -b feature-estilos main
```

```

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb/laboweb (main)
$ git checkout -b feature-estilos main
Switched to a new branch 'feature-estilos'

```

- Modifica el archivo `estilos.css`:

- propiedad color del `body` y de propiedad `h2`: `#2a2a2a`
- `background-color` de `header` y `footer`: `#2a75ff`

```

$ git add css/estilos.css
$ git commit -m "cambiados estilos"

```

```

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb/laboweb (feature-estilos)
$ git add css/estilos.css

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/laboweb/laboweb (feature-estilos)
$ git commit -m "cambiados estilos"
[feature-estilos 516ba1c] cambiados estilos
1 file changed, 3 insertions(+), 2 deletions(-)

```

Usando el editor visual studio code cambie los colores a otros.

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (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
```

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (feature-estilos)
$ git status
On branch feature-estilos
nothing to commit, working tree clean
```

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'.

```
$ git checkout main
```

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (feature-estilos)
$ git checkout main
Switched to branch 'main'
M       css/estilos.css
```

```
$ git status
```

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (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:   css/estilos.css
        modified:   index.html
```

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

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git add index.html

alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git commit -m "añadido auto a index.html"
[main 0b46aa1] añadido auto a index.html
1 file changed, 2 insertions(+)
```

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

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git log --oneline --all --graph --decorate
* 0b46aa1 (HEAD -> main) añadido auto a index.html
| * ef3f82a (feature-estilos) Removed texts that need to be added later on
|/
* 3a5fd55 Added README.txt, LICENCE.txt and passwords.txt to .gitignore
* 633903c Commit inicial
```

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

```
$ git checkout main
$ git merge feature-estilos
```

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (feature-estilos)
$ git checkout main
Switched to branch 'main'

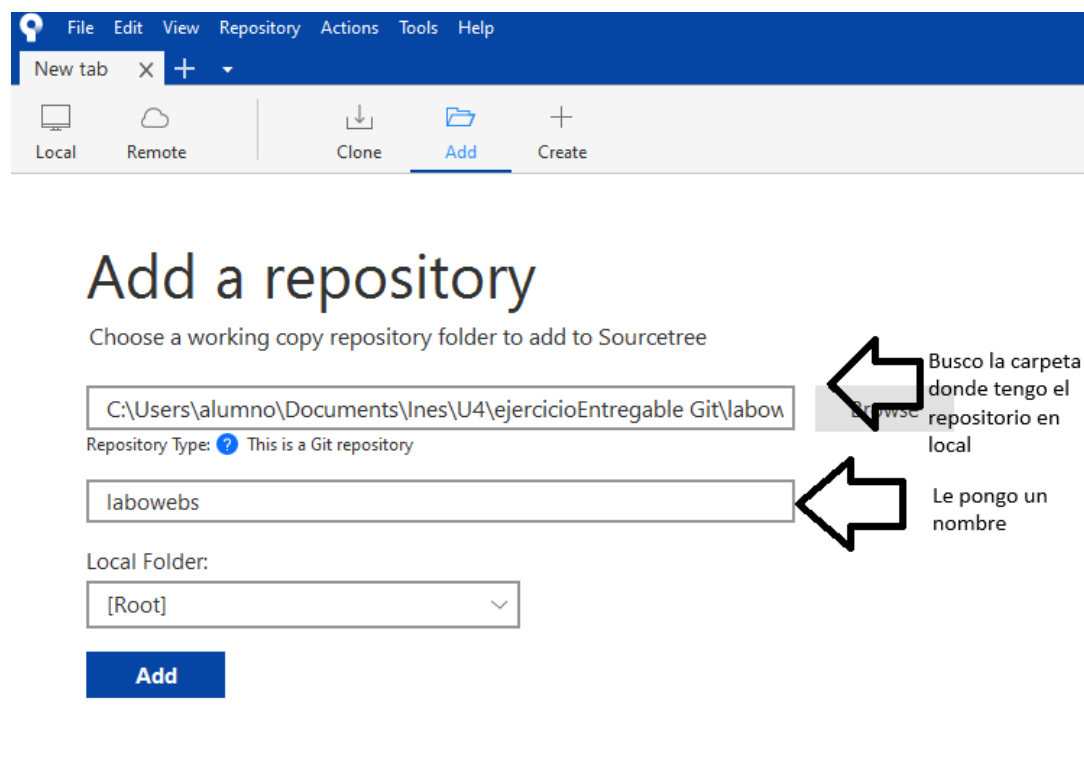
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git merge feature-estilos
Merge made by the 'ort' strategy.
 css/estilos.css | 5 +++--
1 file changed, 3 insertions(+), 2 deletions(-)
```

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

```
alumno@DESKTOP-1H5G46T MINGW64 ~/Documents/Ines/U4/ejercicioEntregable Git/labowebs/labowebs (main)
$ git log --oneline --all --graph --decorate
* ac959ba (HEAD -> main) Merge branch 'feature-estilos'
|
| * 516ba1c (feature-estilos) cambiados estilos
| | b27663a Merge branch 'feature-estilos'
| |
| | * ef3f82a Removed texts that need to be added later on
| | | 0b46aa1 añadido auto a index.html
| |
| |
| * 3a5fd55 Added README.txt, LICENCE.txt and passwords.txt to .gitignore
* 633903c Commit inicial
```

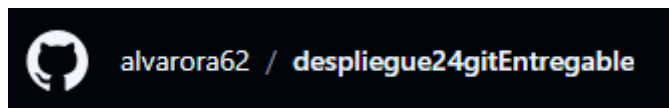
Trabajo en remoto

1. Continúa con el repositorio `labowebs`. Añade el repositorio a Sourcetree.



2. Crea un repositorio remoto y sube al remoto los ficheros de tu repositorio local. Debes subir todas las ramas.

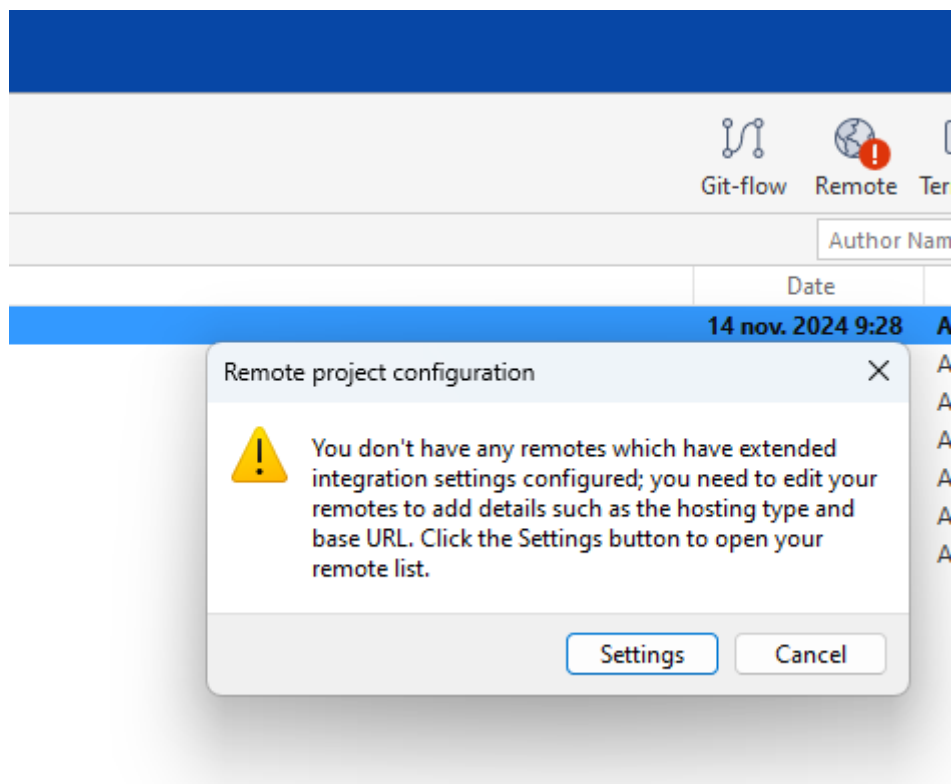
He creado un repositorio publico en git.



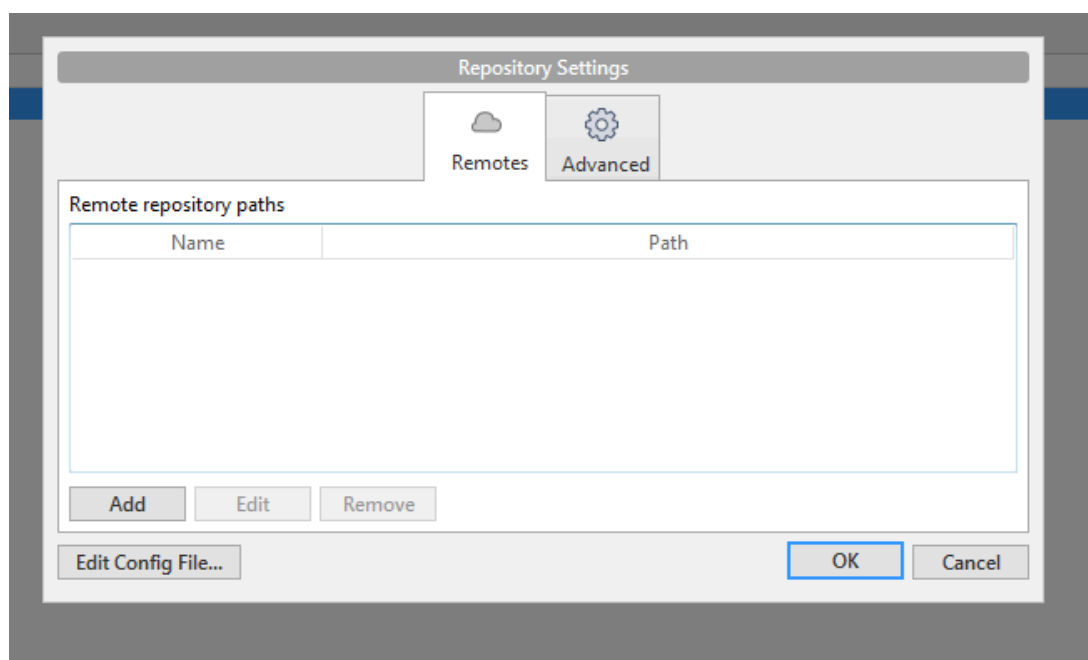
```
$ git remote add origin  
https://github.com/alvarora62/despliegue24gitEntregable.git  
$ git branch -M main  
$ git push -u origin main
```

En Source Tree:

- 1º. Hacemos click sobre "Remote" y en el pop-up que nos muestra en "Settings".



2º.En la nueva ventana que sale le damos a "add".



3º.Le ponemos un nombre al remoto, el que se quiera, en mi caso lo llame "origin", y el enlace del repositorio. Más abajo ponemos el nombre de usuario de git, en mi caso "alvarora62". Una vez rellenado esto, le damos a "OK".

Remote details


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.

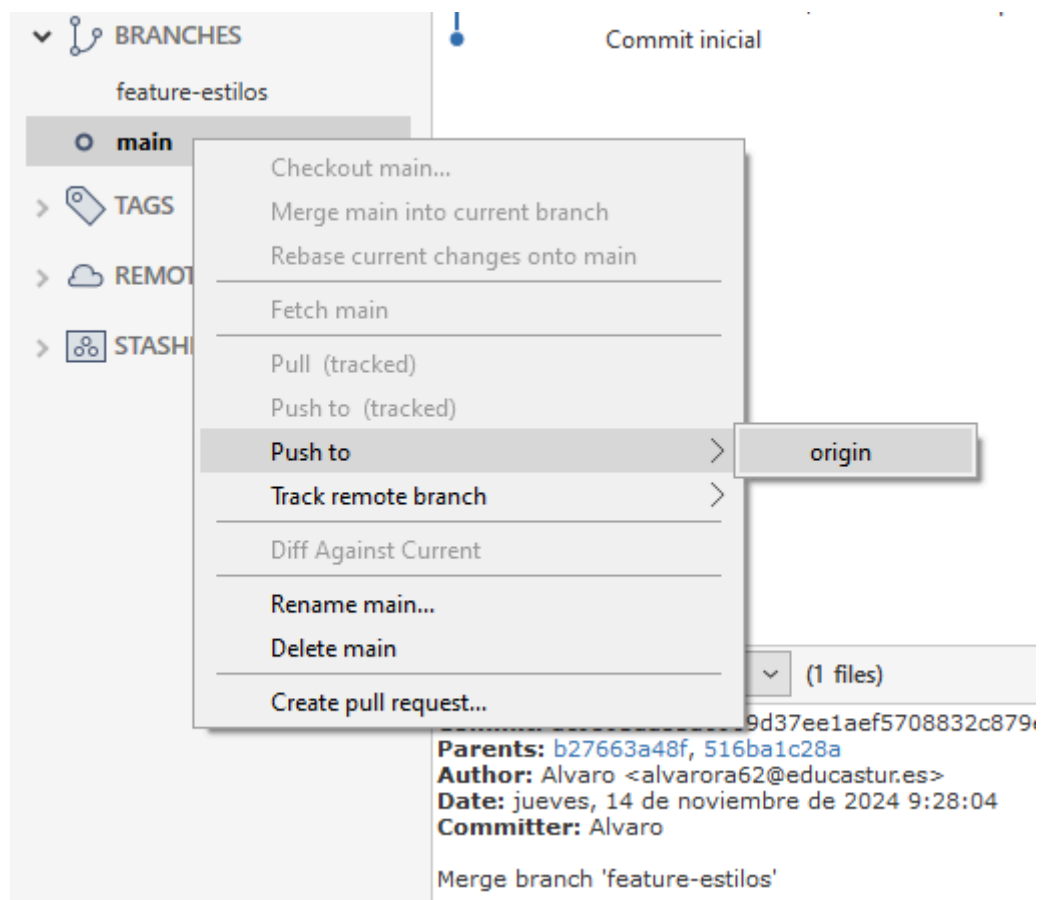
4°.Una vez añadido el remoto le damos a "OK".

Repository Settings

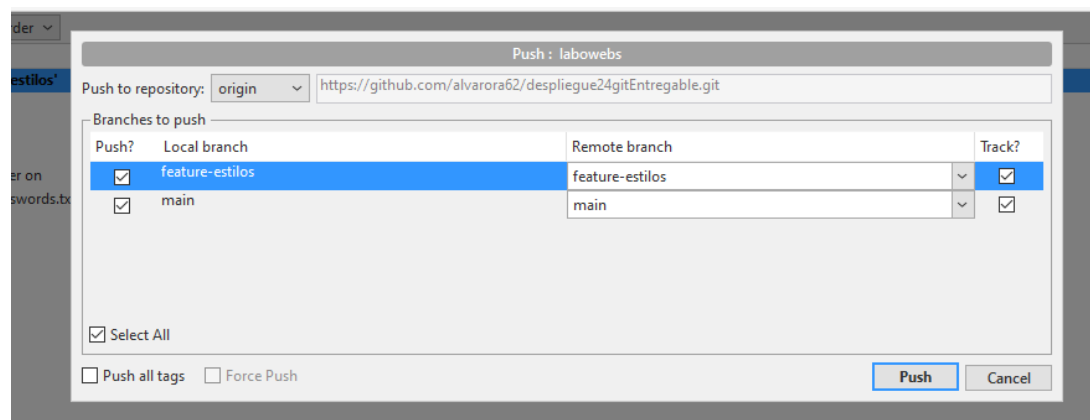
Remote repository paths

Name	Path
origin	https://github.com/alvarora62/despliegue24gitEntregable.git

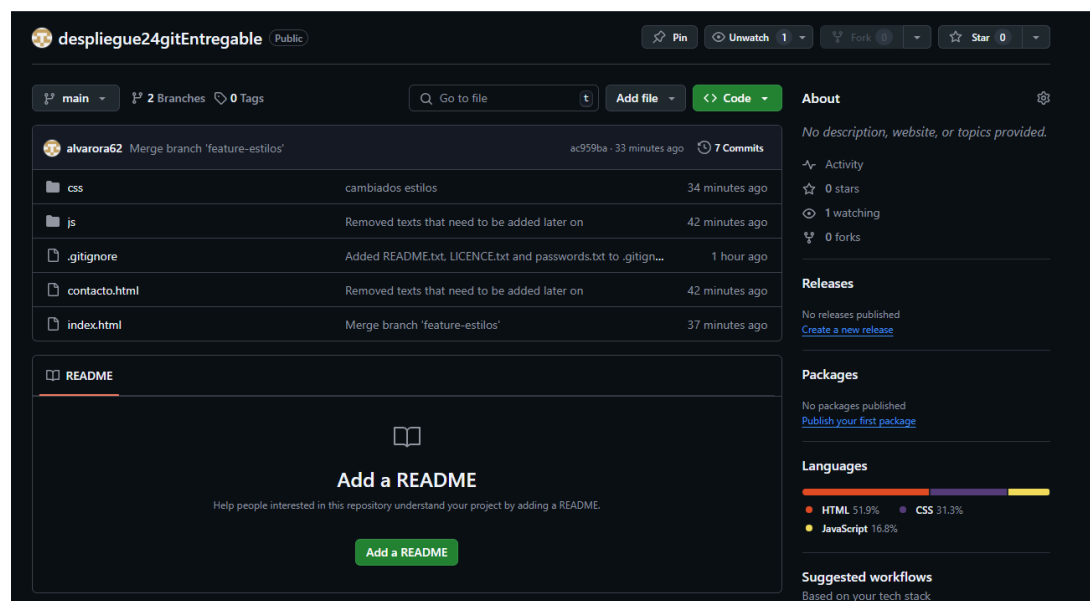
5°.Ahora vamos a hacer el primer push hacia el repositorio, vamos a la barra lateral izquierda, click derecho sobre "main" --> "push to" --> "origin"(o el nombre que se le haya puesto al remoto)



6°.En mi caso quiero subir ambas ramas, así que marco todas las casillas. Termino dándole a "push".



7°.Ahora ya lo tengo subido a github.

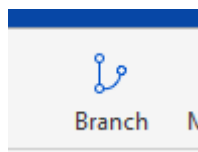


3. **Crea una rama** `feature-index`. **Añade el siguiente código dentro de la** `<section class="about">`. **Añade los cambios y crea un commit. Sube los cambios al remoto.**

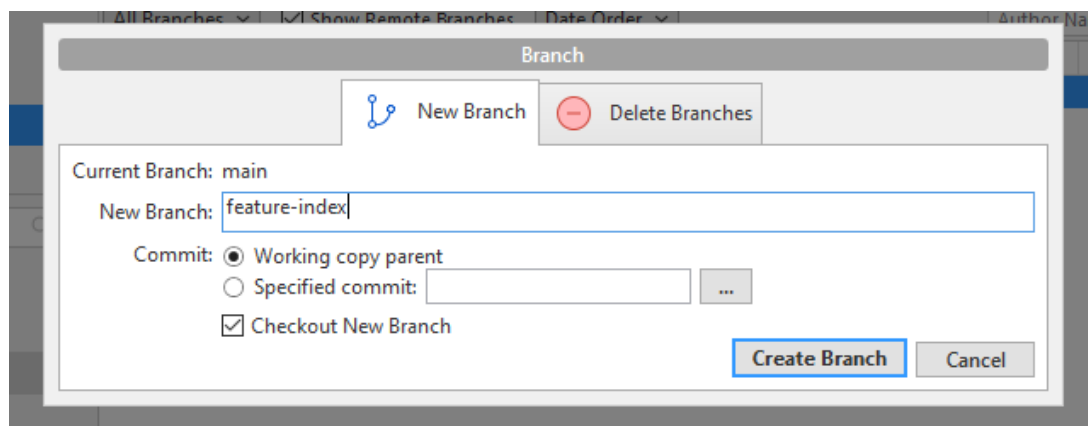
El código a añadir:

```
<h2>Conoce a Nuestro Equipo</h2>
<p>Labowebs está formado por un equipo de diseñadores y
desarrolladores apasionados que trabajan juntos para ofrecer soluciones
tecnológicas de alta calidad. Cada miembro de nuestro equipo aporta
experiencia en diseño, programación, y soporte técnico, asegurando que
nuestros clientes reciban un servicio completo y personalizado.</p>
<p>Nuestro objetivo es que cada cliente se sienta acompañado en su
aventura digital, con un equipo profesional que entiende sus necesidades
y
trabaja para hacer crecer su presencia en línea.</p>
```

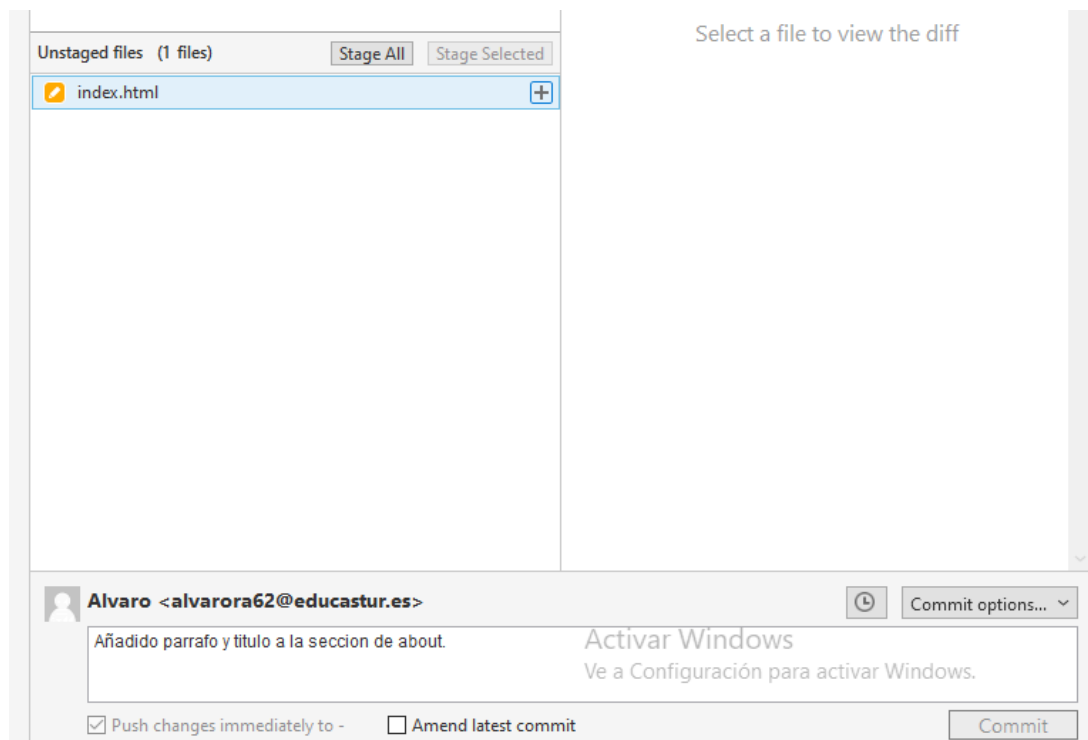
Damos click encima del botón que pone "branch".



En esta nueva ventana ponemos el nombre de la rama nueva.

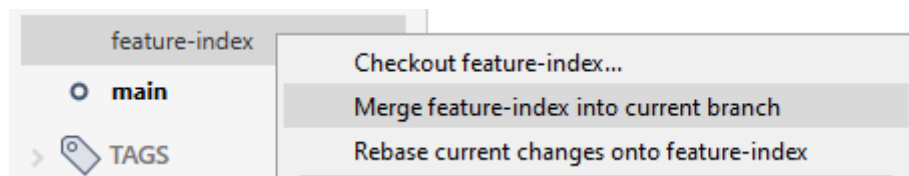


Una vez realizado los cambio sobre el fichero `index.html` le damos al + para añadirlo al commit y hacemos un commit & push marcando la casilla.



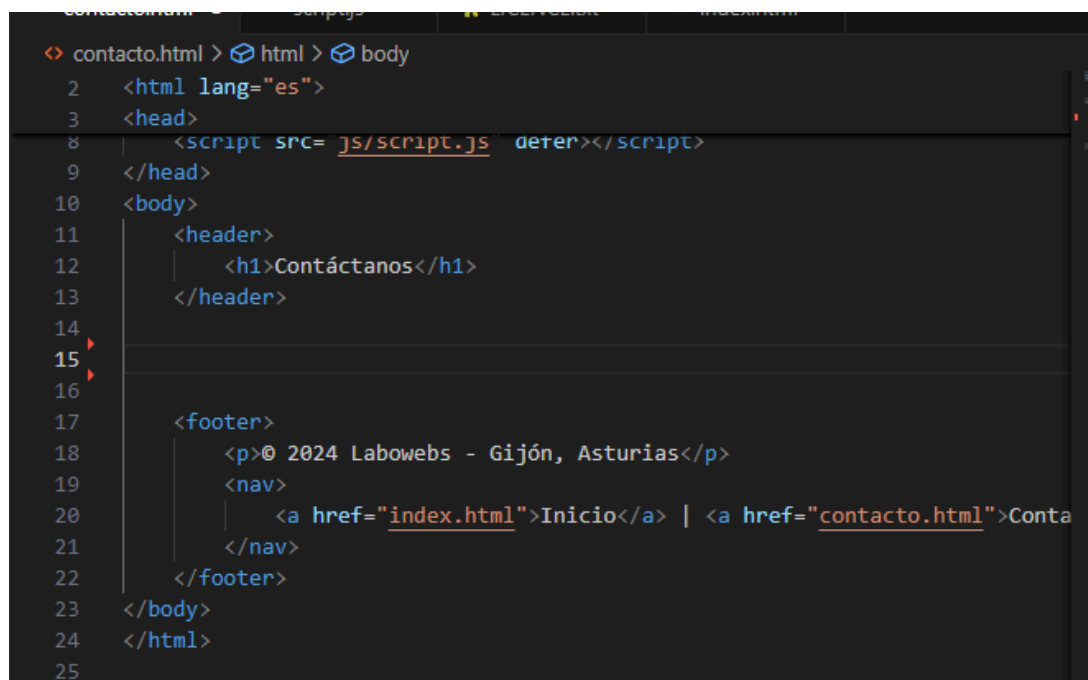
4. En el repositorio local, fusiona la rama `feature-index` en la rama `main`.

Nos vamos a la rama `main` haciendo doble click sobre ella. Después le damos click derecho a la rama `feature-index` y pulsamos merge.

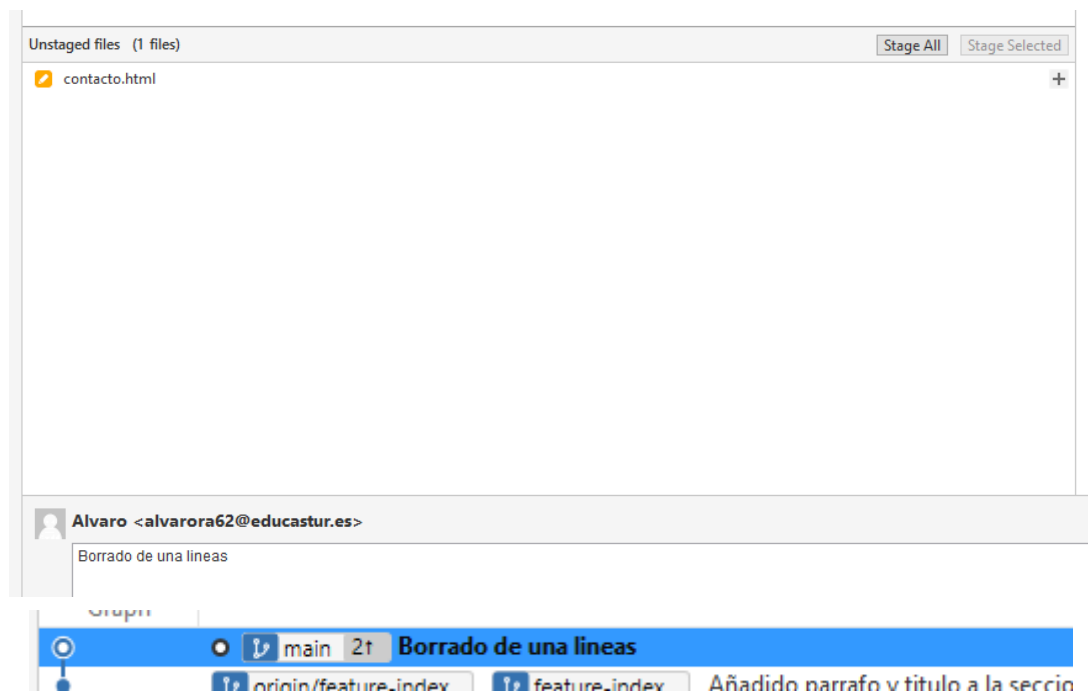


5. Edita el fichero `contacto.html`. Borra unas líneas. Muestra los ficheros con cambios pendientes y las diferencias. Añade los cambios y haz un commit.

Eliminamos una sección dentro de `contacto.html`.

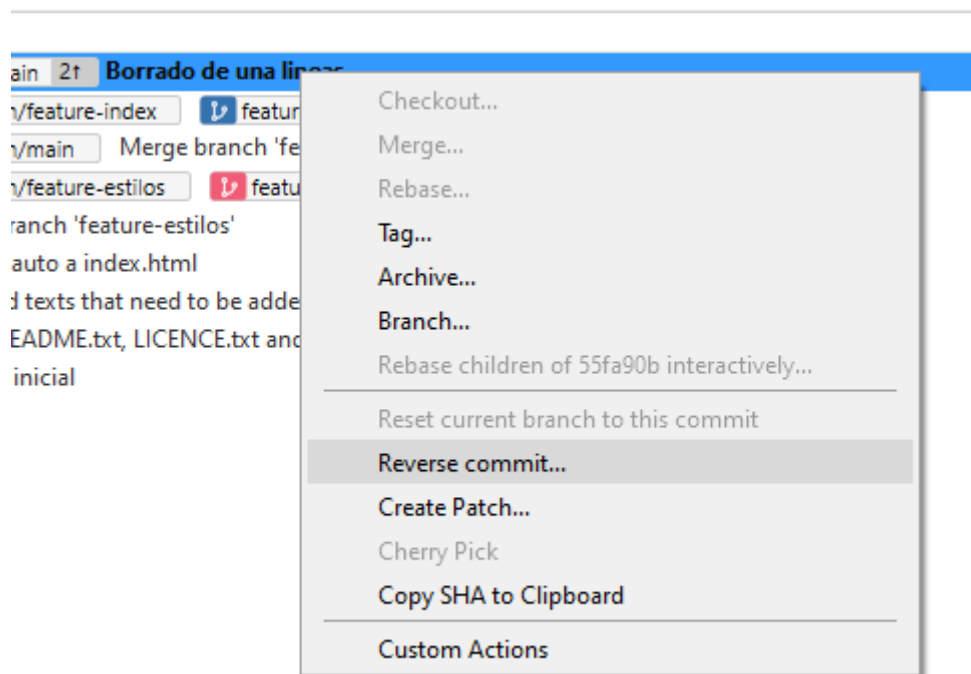


En el apartado de "File Status" marcamos el "+" para añadir el archivo al commit, escribimos un mensaje de commit y hacemos commit.



6. Te das cuenta del error. Deshaz el commit anterior. Captura el estado actual del repositorio.

Buscamos en "History" el ultimo commit realizado sobre el borrado, le damos click derecho encima de él y pulsamos "Reverse commit...", luego en la ventana que nos sale le damos a "Yes".



Aqui podemos ver que se ha revertido y si miramos de nuevo el archivo veremos que esas lineas que borramos vuelven a estar.



```

2 <html lang="es">
3 <head>
8 <script src="js/script.js" defer></script>
9 </head>
10 <body>
11 <header>
12 <h1>Contáctanos</h1>
13 </header>
14
15 <section class="contact-form">
16 <h2>Envíanos un mensaje</h2>
17 <form id="formContacto" action="#" method="post">
18 <label for="nombre">Nombre:</label>
19 <input type="text" id="nombre" name="nombre" required>
20
21 <label for="email">Correo Electrónico:</label>
22 <input type="email" id="email" name="email" required>
23
24 <label for="mensaje">Mensaje:</label>
25 <textarea id="mensaje" name="mensaje" rows="4" required></textare
26
27 <button type="submit">Enviar</button>
28 </form>
29 </section>
30
31 <footer>
32 <p>© 2024 Labowebs - Gijón, Asturias</p>
33 <nav>
34 <a href="index.html">Inicio</a> | <a href="contacto.html">Conta
35 </nav>
36 </footer>
37 </body>
38 </html>
39

```

7. Crea una rama `feature-mapa`. Incluye este código en el archivo `contacto.html`.
 Añade los cambios. Realiza un commit. Sube los cambios al remoto. Muestra en el
 remoto los cambios del archivo `contacto.html` en la rama `feature-mapa`.

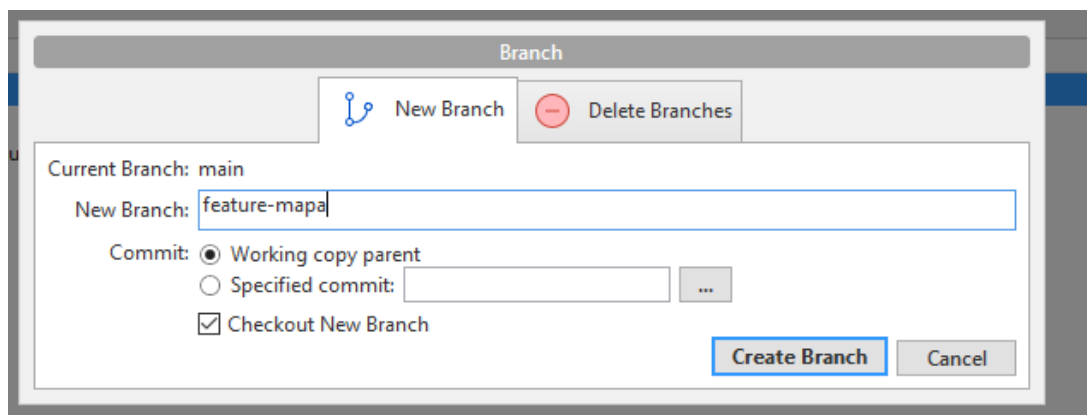
El código a añadir:

```

<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>

```

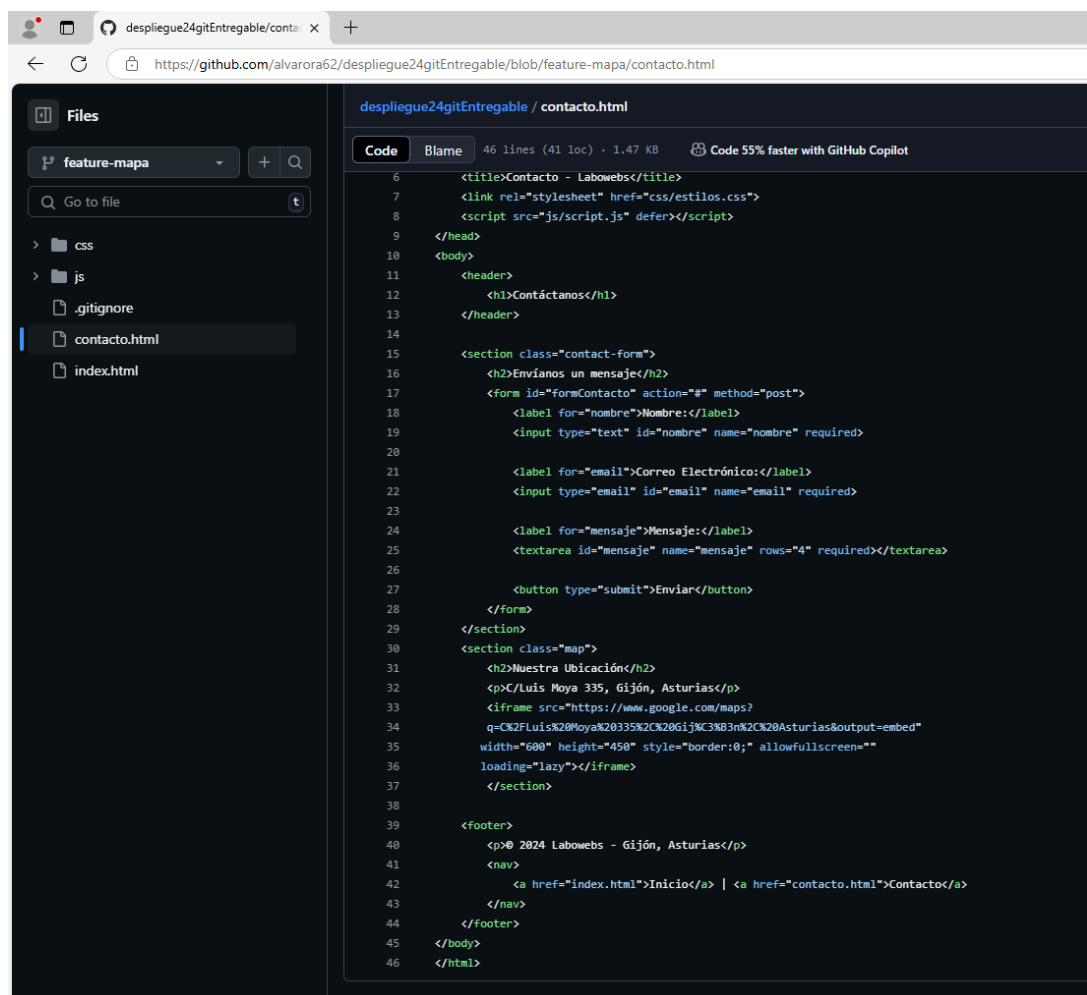
Creamos una nueva rama llamada "feature-mapa":



Añadimos el código al archivo y hacemos un commit & push marcando la casilla.

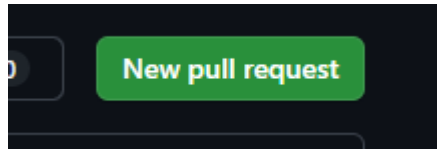
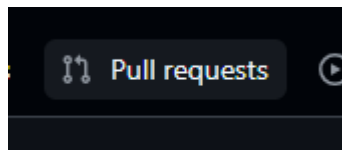


Aquí podemos ver desde github el cambio que hemos realizado.

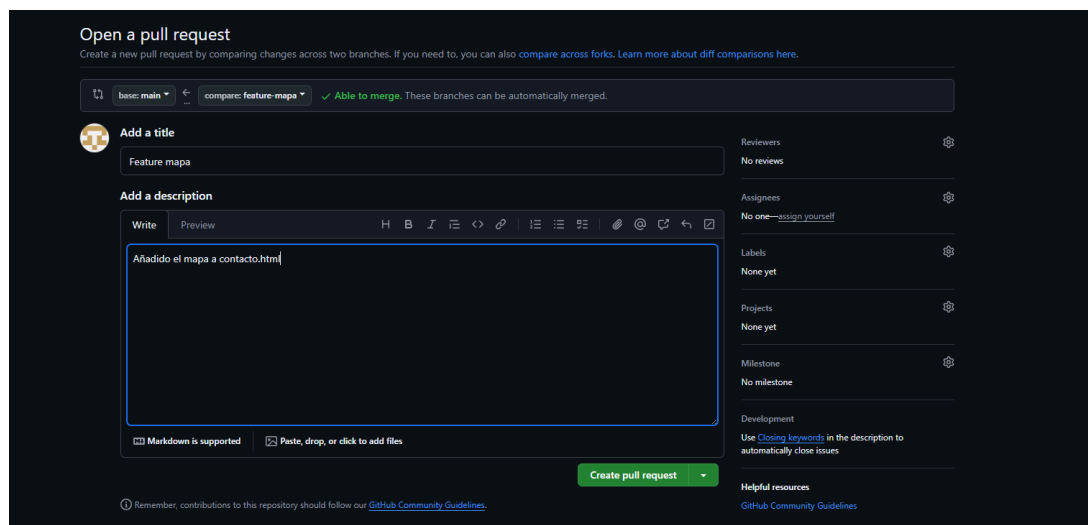
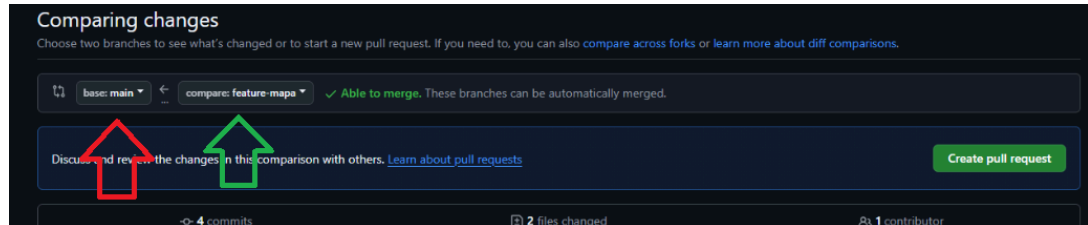


- En GitHub, en la rama `main`, fusiona la rama `feature-mapa`. Baja los cambios del remoto a local. Deja los dos repositorios sincronizados

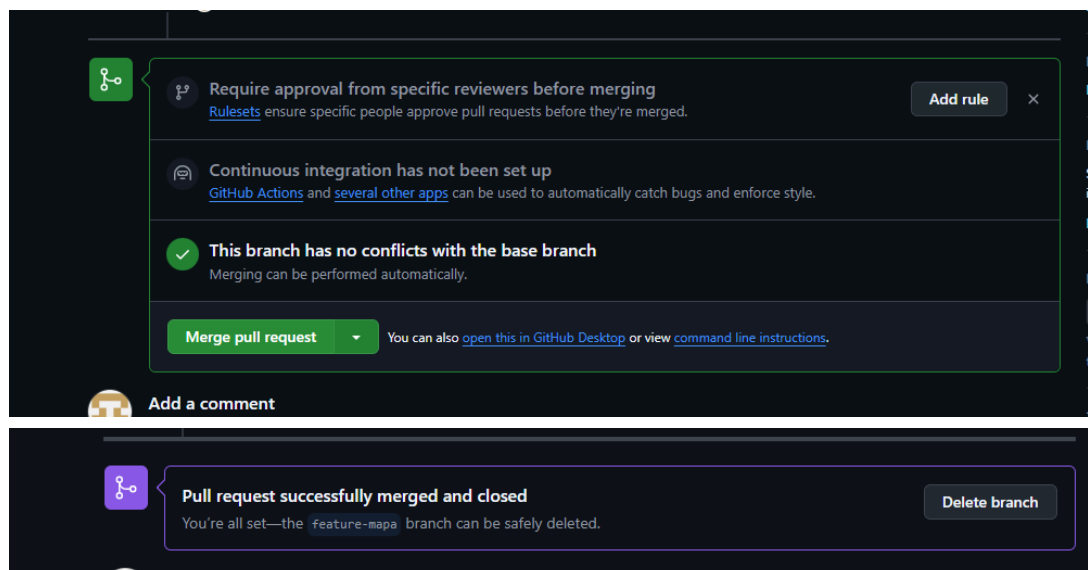
Para sincronizar los cambios desde github tendremos que hacer una pull request, para ello buscamos en el menu "Pull requests" y le damos click. Despues al botón que dice "New pull request".



Nos saldrán estas opciones, en "base" ponemos la rama objetivo, donde queremos unir todo, y en "compare" la rama que contiene esa funcionalidad nueva o arreglo que queremos mergear. Una vez seleccionado abajo nos saldrá la lista con todos los cambios. Una vez hecho le damos a "Create pull request".



Una vez creada la pull request al pulsar en "Merge pull request" se habrán mergeado las ramas.



Para actualizar ahora el local tendremos que hacer un pull de la rama main.

ido

o par

ados

gitig

Pull

Pull from remote: origin

https://github.com/alvarora62/despliegue24gitEntregable.git

Remote branch to pull: main

Refresh

Pull into local branch: feature-mapa

Options

☒ Commit merged changes immediately

☐ Include messages from commits being merged in merge commit

☐ Create a new commit even if fast-forward is possible

☐ Rebase instead of merge (WARNING: make sure you haven't pushed your changes)

Pull

Cancel

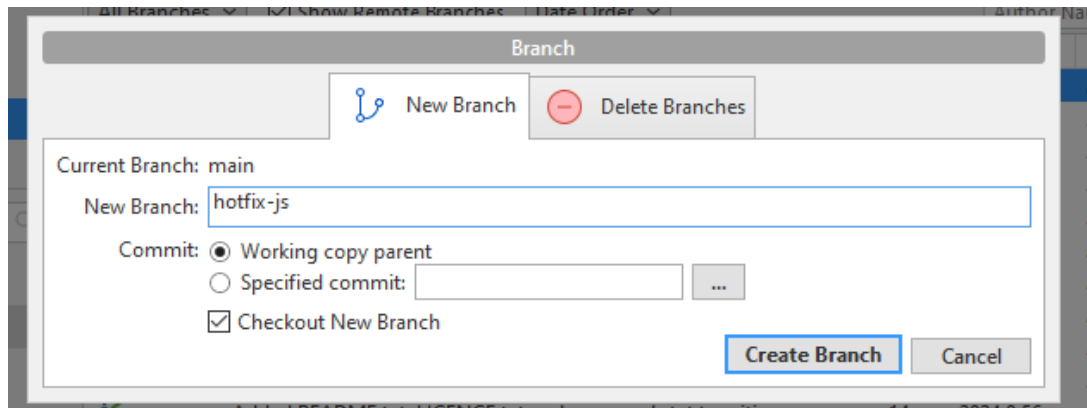
Conflictos

1. **Crea una rama `hotfix-js`. Cámbiate a ella. Añade este código en el fichero `script.js`. Confirma el cambio y haz un commit. (Fíjate en los números de línea...)**

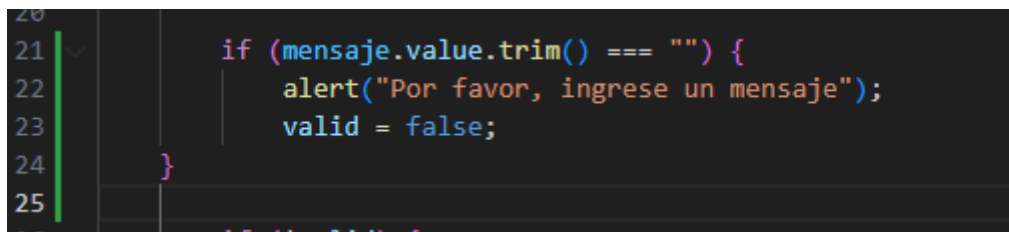
El código a añadir:

```
if (mensaje.value.trim() === "") {  
    alert("Por favor, ingrese un mensaje");  
    valid = false;  
}
```

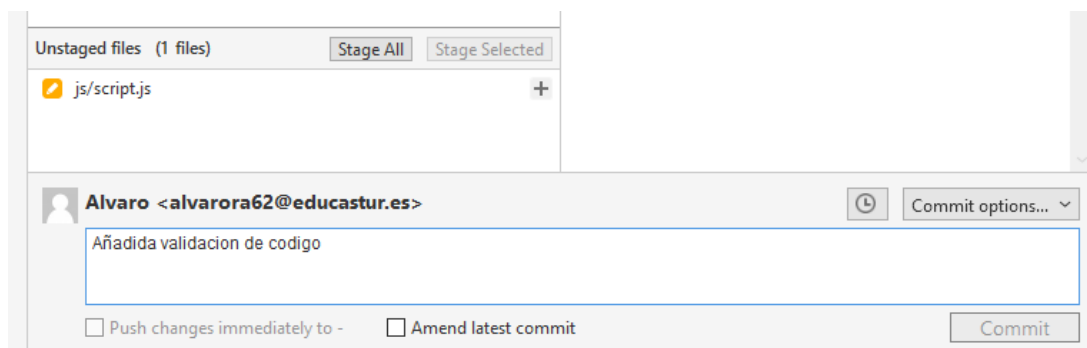
Creo una nueva rama con el nombre especificado y me sitúo en ella.



Después, abro Visual Studio Code para editar el archivo `scrip.js`.



Confirmo el cambio y hago el commit.

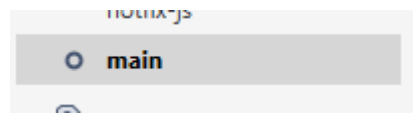


2. **Vuelve a la rama `main`. En el fichero `script.js` en las mismas líneas que en la cuestión anterior, añade el código siguiente. Confirma el cambio y haz un commit.**

El código a añadir:

```
if (mensaje.value.trim() === "") {  
    alert("Ingrese un mensaje, por favor");  
    valid = false;  
}
```

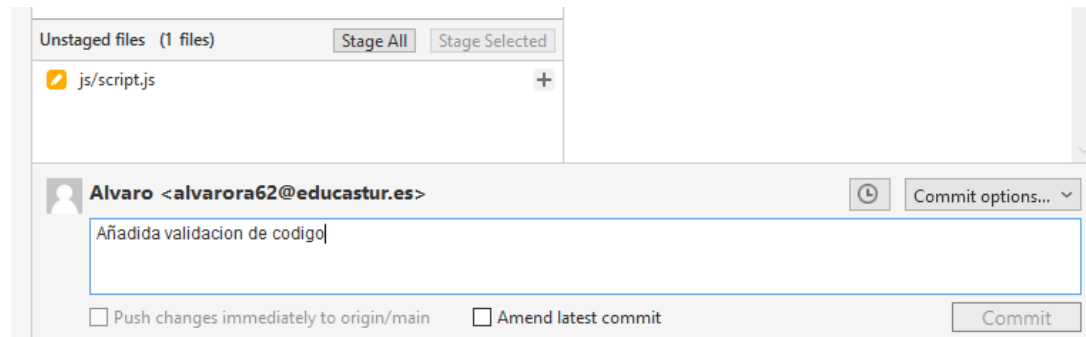
Me voy a la rama `main` haciendo doble click sobre ella.



Hago el cambio en `script.js`.

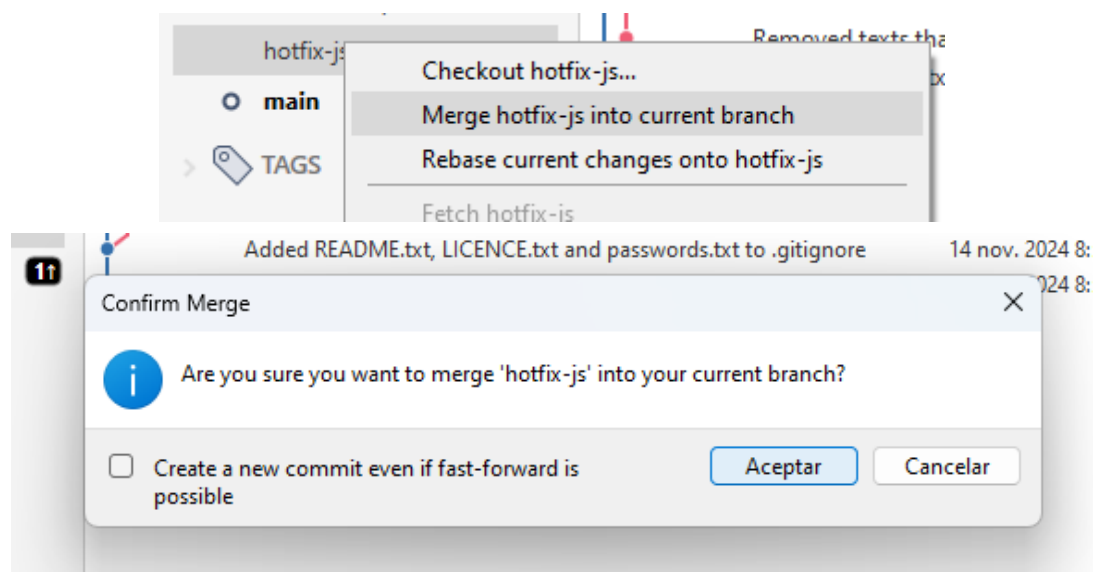
```
20
21     if (mensaje.value.trim() === "") {
22         alert("Ingrese un mensaje, por favor");
23         valid = false;
24     }
25
```

Le hago confirmo el cambio y hago un commit.

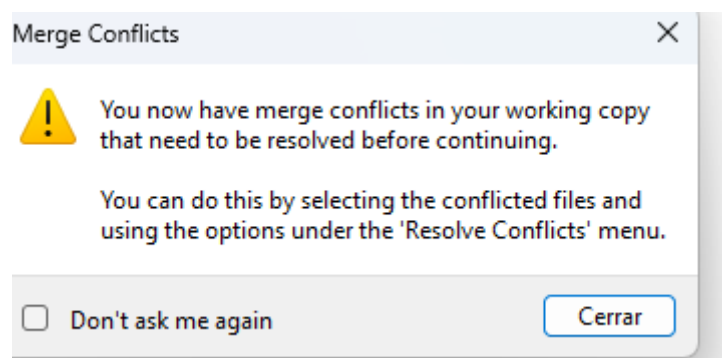


3. **Fusiona la rama `hotfix-js` en `main`. Debe producirse un conflicto. Resuélvelo. Cuando termines la resolución del conflicto sube los cambios al remoto - Deja los repositorios sincronizados.**

Me voy a la rama `main` y hago click derecho encima de la rama `hotfix-js`. En la ventana que sale le digo que si.



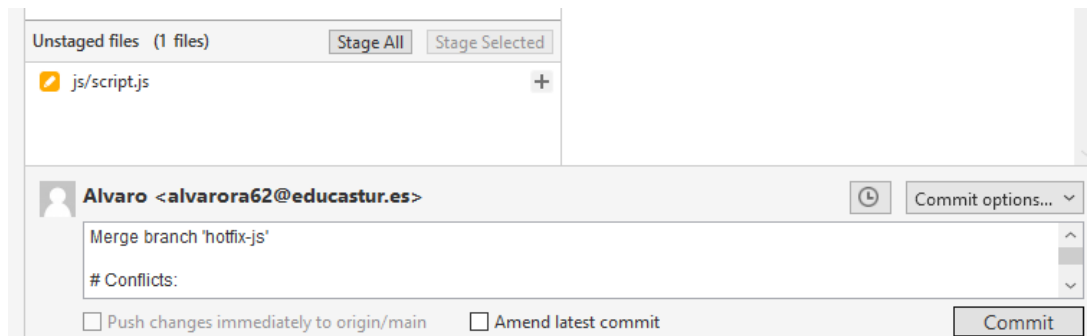
Me muestra que hay conflictos con el merge que estoy haciendo.



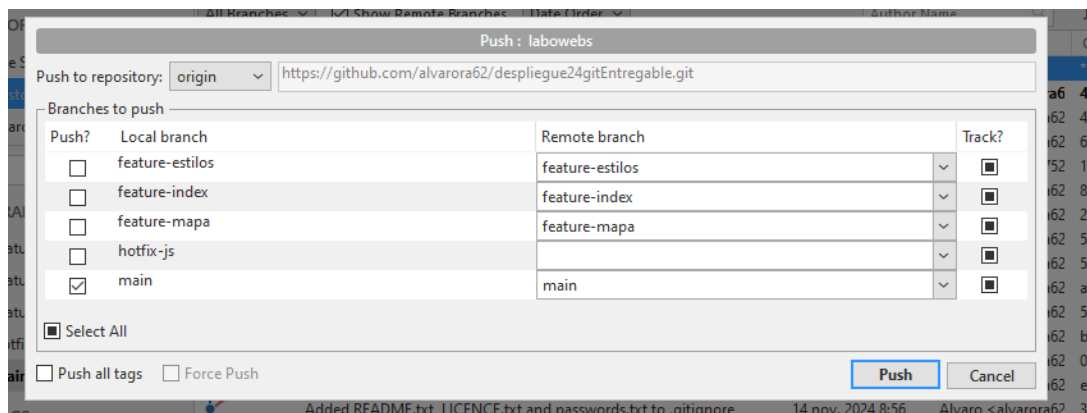
Desde Visual Studio Code puedo ver y solucionar los conflictos.

```
18     valid = false;
19   }
20
21   if (mensaje.value.trim() === "") {
22     <<<<<< HEAD (Current Change)
23     alert("Ingrese un mensaje, por favor");
24     =====
25     alert("Por favor, ingrese un mensaje");
26     >>>>>> hotfix-js (Incoming Change)
27     valid = false;
28   }
29
30   if (!valid) {
31     event.preventDefault();
32   }
```

Una vez solucionados los conflictos hago un commit y ya estaría el merge.



Finalmente lo subo al repositorio en github.



Extra

- Enlace al repositorio de github: [click](#)