

TAREA 2. EJERCICIO DE GIT - PROYECTO LABDIST

TRABAJO EN LOCAL

TRABAJO EN REMOTO

CONFLICTOS

TAREA 2. EJERCICIO DE GIT - PROYECTO LABDIST

Autor: Jesús Alberto López Vázquez

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.

```
mkdir labdist -> Para crear el fichero labdist  
cd labdist -> Colocarnos en el fichero creado  
git init -> Para crear el repositorio del fichero  
git branch -m main -> Renombrar la rama master por main  
git add . -> para agregar los archivos al staging area  
git status -> mostrar el estado actual del repositorio  
git commit -m "mensaje" -> realizar el commit aplicandole un  
mensaje
```

```
MINGW64:/d:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2
$ mkdir labdist

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2
$ git init labdist
Initialized empty Git repository in D:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist/.git/

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2
$ cd labdist

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
master)
$ git branch -m main

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$
```

```
MINGW64:/d:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git add .

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git status
On branch main

No commits yet

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

User@DESKTOP-DMGAIPO MINGW64 /d:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$
```

```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist

User@DESKTOP-DMGAIPO MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git commit -m "commit inicial"
[main (root-commit) 21f2071] commit inicial
5 files changed, 198 insertions(+)
create mode 100644 .gitignore
create mode 100644 contacto.html
create mode 100644 css/estilos.css
create mode 100644 index.html
create mode 100644 js/script.js

User@DESKTOP-DMGAIPO MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$
```

```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist

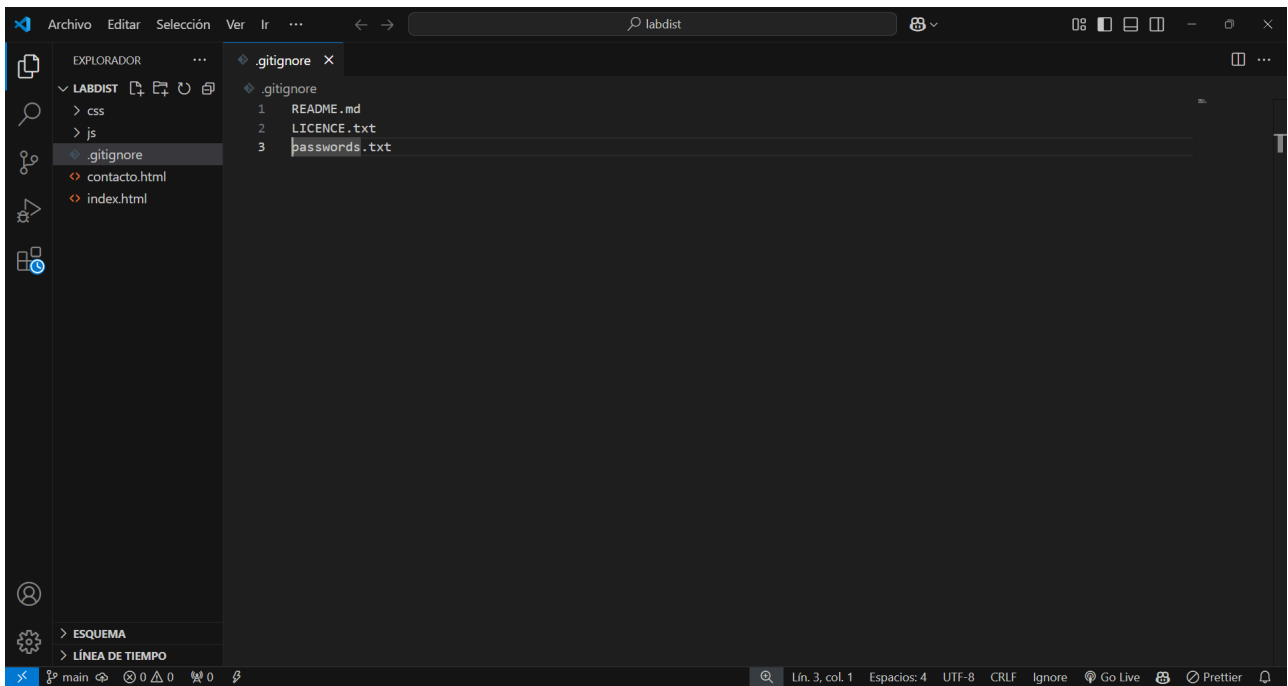
User@DESKTOP-DMGAIPO MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git log
commit 21f2071ce82b2b5e29d5fb8e5f3b57a71607f2af (HEAD -> main, feature-estilos)
Author: JesusAlumnoDAW <jalberto1v16@educastur.es>
Date: Thu Jan 30 12:59:52 2025 +0100

    commit inicial

User@DESKTOP-DMGAIPO MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/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.

```
git log --oneline -> mostrar los logs del repositorio en una línea
```



```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git log
commit 21f2071ce82b2b5e29d5fb8e5f3b57a71607f2af (HEAD -> main, feature-estilos)
Author: JesusAlumnoDAW <jalbertolv16@educastur.es>
Date: Thu Jan 30 12:59:52 2025 +0100

    commit inicial

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git log --oneline
21f2071 (HEAD -> main, feature-estilos) commit inicial

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/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.

```
echo "mensaje > README.md" -> escribir mensaje en README.md
echo "mensaje" LICENCE.txt -> Dar contenido a LICENCE.txt
echo "mensaje" passwords.txt -> escribir contenido en passwords.txt
git status -> ver el estado del repositorio
git status --ignored -> ver archivos ignorados
```

```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist
Date: Thu Jan 30 12:59:52 2025 +0100

commit inicial

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git log --oneline
21f2071 (HEAD -> main, feature-estilos) commit inicial

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ echo "Archivo README" > README.md

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ echo "Licencia proyecto" > LICENCE.txt

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ echo "contraseña123" > passwords.txt

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$
```

```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist
main)
$ echo "contraseña123" > passwords.txt

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git status
On branch main
nothing to commit, working tree clean

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/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

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ |
```

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

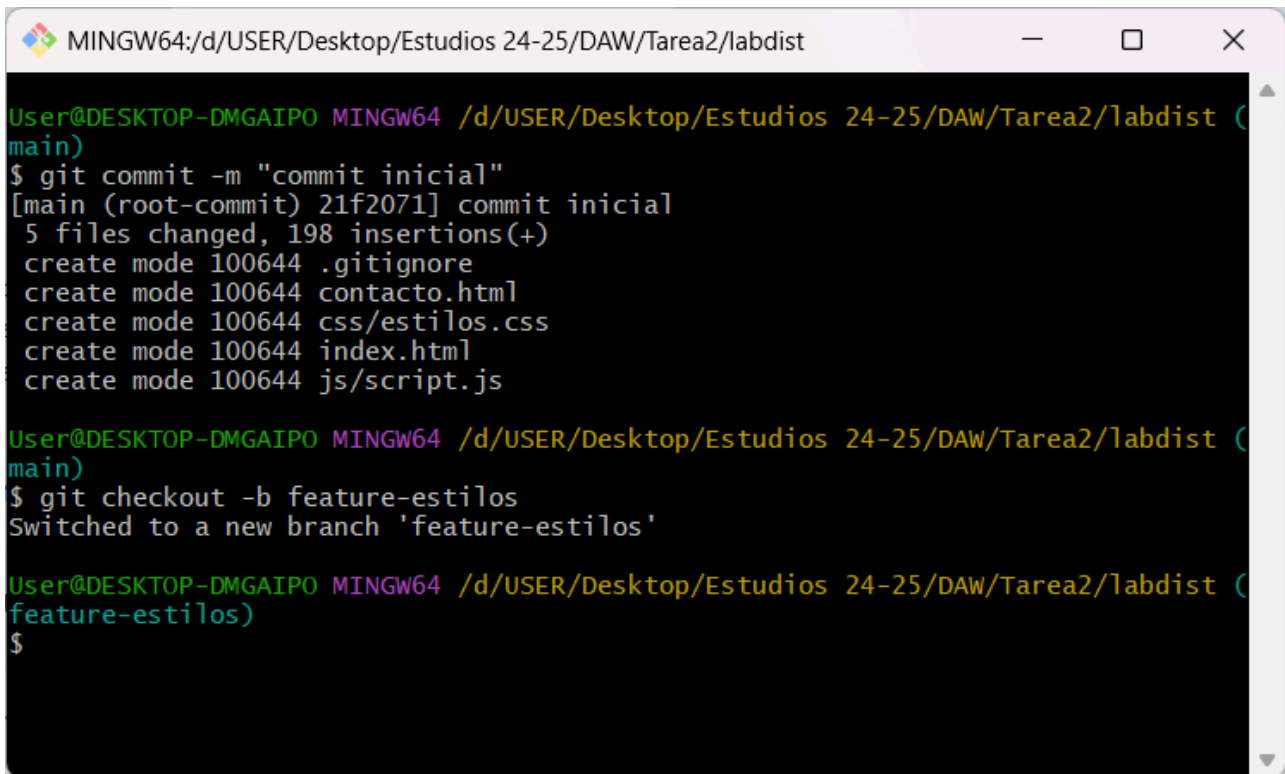
Modifica el archivo estilos.css :

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"

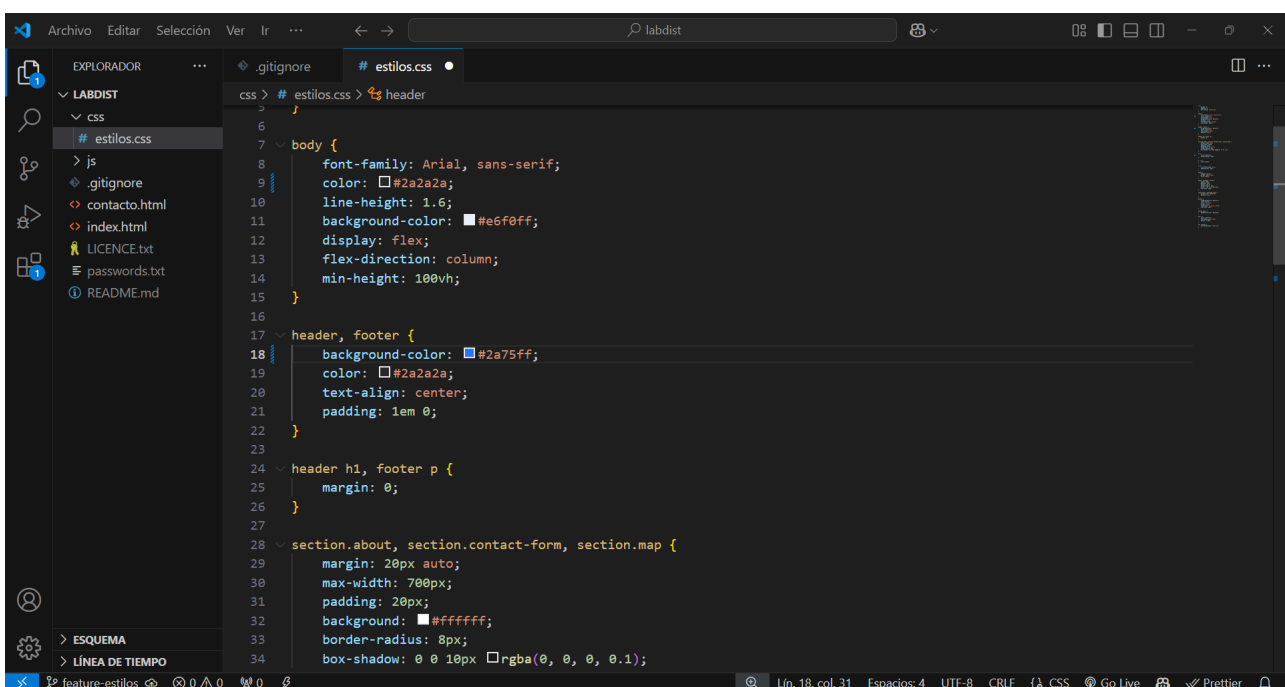
```
git checkout -b feature-estilos --> Crea la rama feature-estilos y
cambia a esa rama
git status -> comprobar el estado del repositorio
git add css/estilos.css -> añadir a staging area los cambios
git commit -m -> confirmar los cambios con un commit, escribiendo
un mensaje que los defina
```



```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist
User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (main)
$ git commit -m "commit inicial"
[main (root-commit) 21f2071] commit inicial
5 files changed, 198 insertions(+)
create mode 100644 .gitignore
create mode 100644 contacto.html
create mode 100644 css/estilos.css
create mode 100644 index.html
create mode 100644 js/script.js

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (main)
$ git checkout -b feature-estilos
Switched to a new branch 'feature-estilos'

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (feature-estilos)
$
```



```
EXPLORADOR
LABDIST
css
# estilos.css
js
.gitignore
contacto.html
index.html
LICENCE.txt
passwords.txt
README.md
ESQUEMA
LÍNEA DE TIEMPO

css > # estilos.css > header
6
7 < body {
8     font-family: Arial, sans-serif;
9     color: #2a2a2a;
10    line-height: 1.6;
11    background-color: #e6f0ff;
12    display: flex;
13    flex-direction: column;
14    min-height: 100vh;
15 }
16
17 < header, footer {
18     background-color: #2a75ff;
19     color: #2a2a2a;
20     text-align: center;
21     padding: 1em 0;
22 }
23
24 < header h1, footer p {
25     margin: 0;
26 }
27
28 < section.about, section.contact-form, section.map {
29     margin: 20px auto;
30     max-width: 700px;
31     padding: 20px;
32     background: #ffffff;
33     border-radius: 8px;
34     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
35 }
```

```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/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")

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
feature-estilos)
$ .....
```

```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist

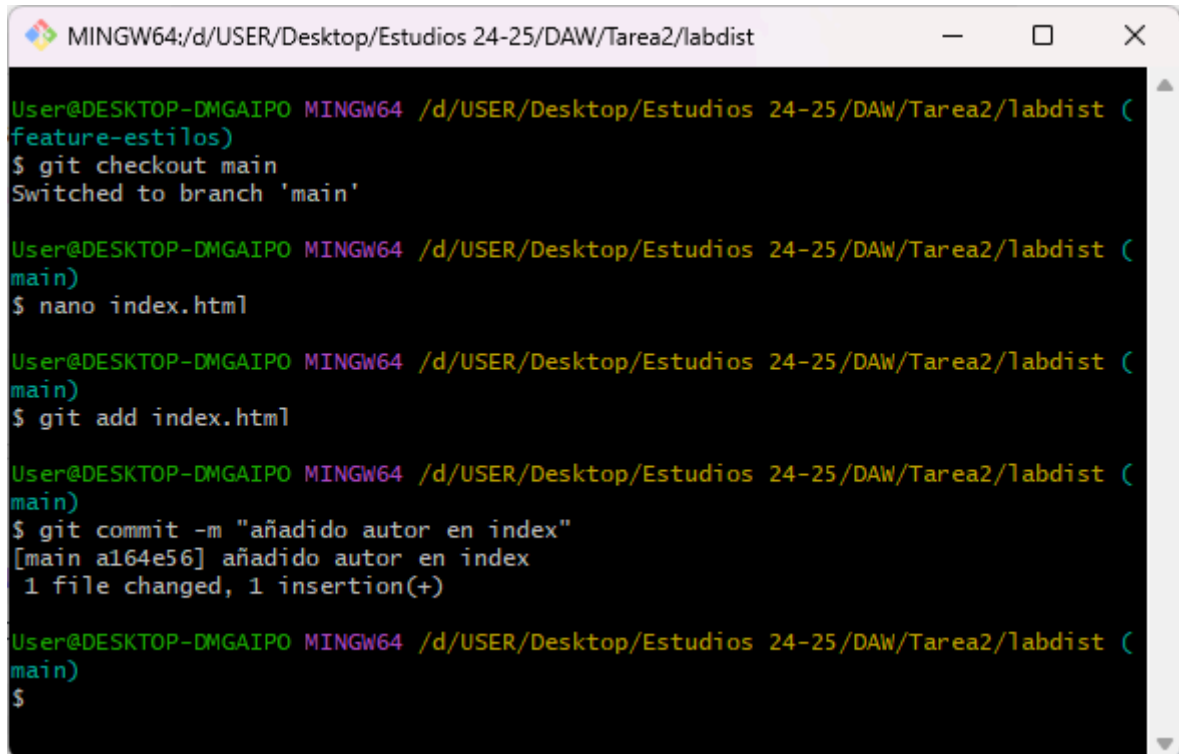
User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
feature-estilos)
$ git add css/estilos.css

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
feature-estilos)
$ git commit -m "actualizados estilos azules"
[feature-estilos 636aa0c] actualizados estilos azules
1 file changed, 3 insertions(+), 3 deletions(-)

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/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'

```
git checkout main -> volver a la rama main
nano index.html -> para modificar el archivo de texto index.html
git add index.html -> enviar cambios al staging area
git commit -m -> hacer el commit dejando mensaje de definicion del
cambio
git log --oneline --graph --decorate --all -> mostrar los log del
repositorio en una sola linea, con gráficos, decoracion y todos
```

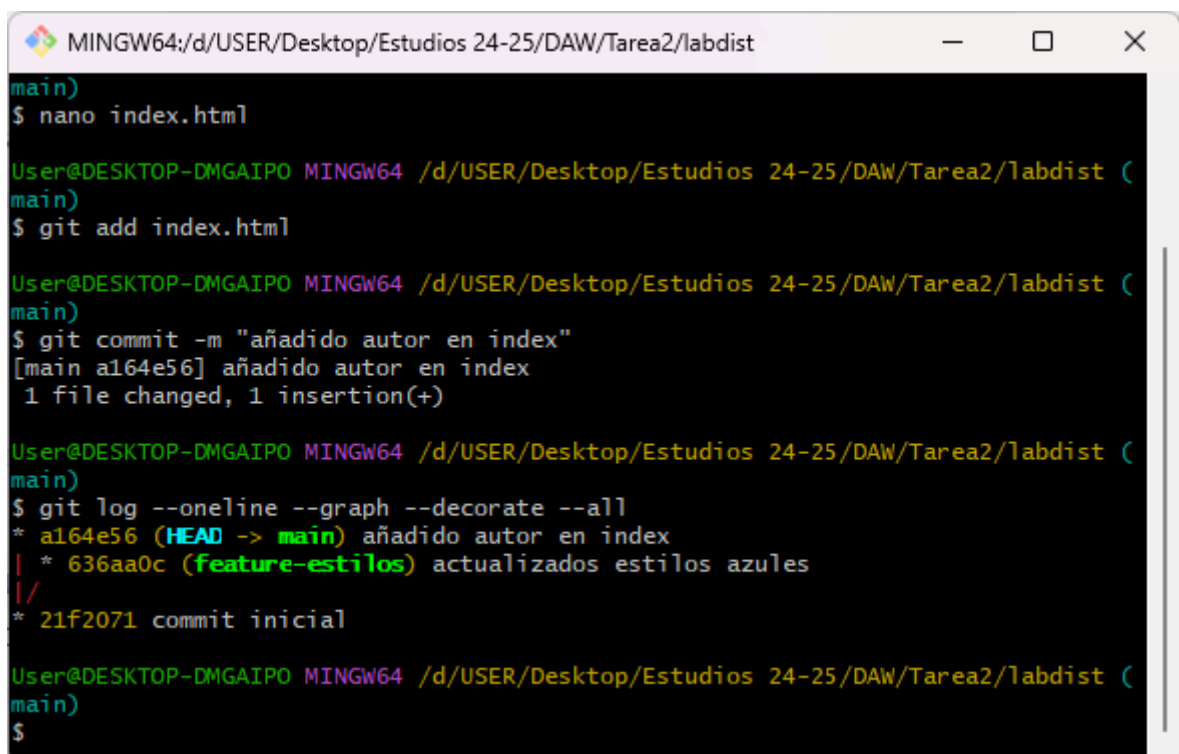


```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist
User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
feature-estilos)
$ git checkout main
Switched to branch 'main'

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ nano index.html

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git add index.html

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git commit -m "añadido autor en index"
[main a164e56] añadido autor en index
1 file changed, 1 insertion(+)
```



```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist
main)
$ nano index.html

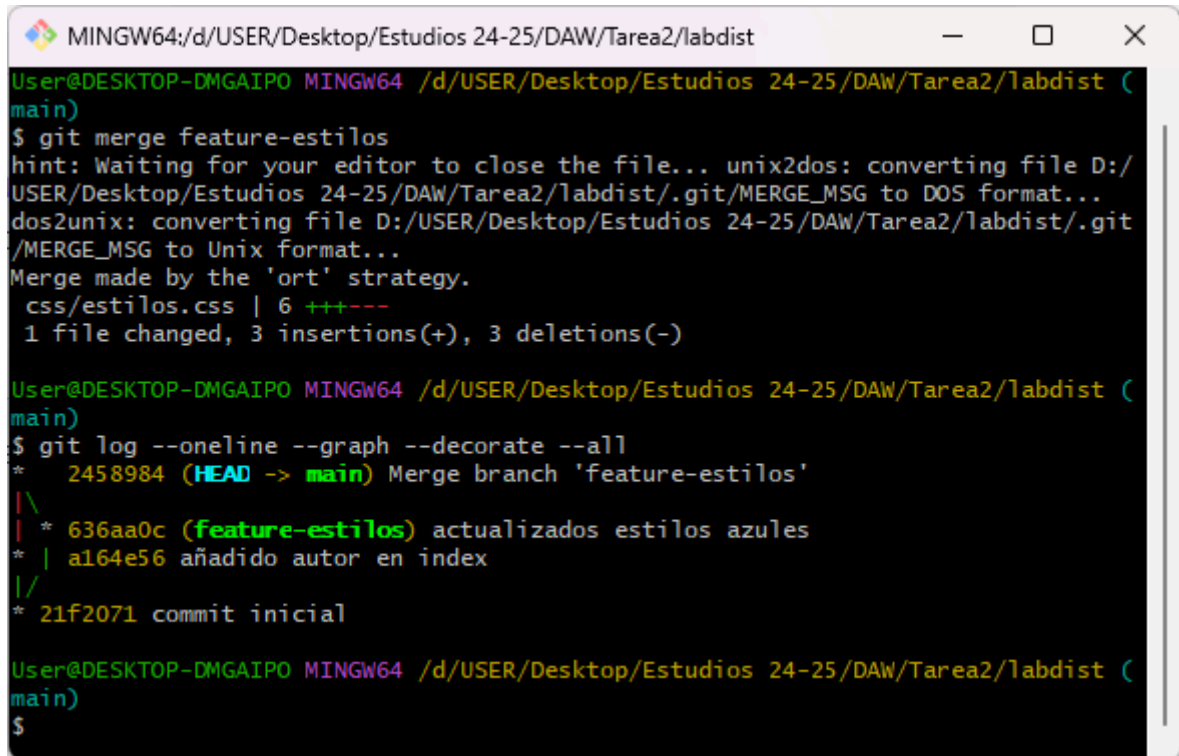
User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git add index.html

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git commit -m "añadido autor en index"
[main a164e56] añadido autor en index
1 file changed, 1 insertion(+)
```

```
User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git log --oneline --graph --decorate --all
* a164e56 (HEAD -> main) añadido autor en index
| * 636aa0c (feature-estilos) actualizados estilos azules
|/
* 21f2071 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 merge feature-estilos -> desde la rama main indico que fusione
feature-estilos a la rama en la que estoy que es la main
git log --oneline --graph --decorate --all -> mostrar los log
segun características indicadas
```

A screenshot of a Windows terminal window titled 'MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist'. The terminal shows the execution of two Git commands. The first command is 'git merge feature-estilos', which results in a successful merge of the 'feature-estilos' branch into the 'main' branch. The output shows the merge strategy used ('ort'), file conversions, and a summary of changes: '1 file changed, 3 insertions(+), 3 deletions(-)'. The second command is 'git log --oneline --graph --decorate --all', which displays a graphical log of the repository's history. The log shows three commits: the initial commit (21f2071), a commit adding the author to the index (a164e56), and a commit updating blue styles (636aa0c). The 'HEAD' is shown pointing to the 'main' branch at commit 2458984, which is the result of the merge.

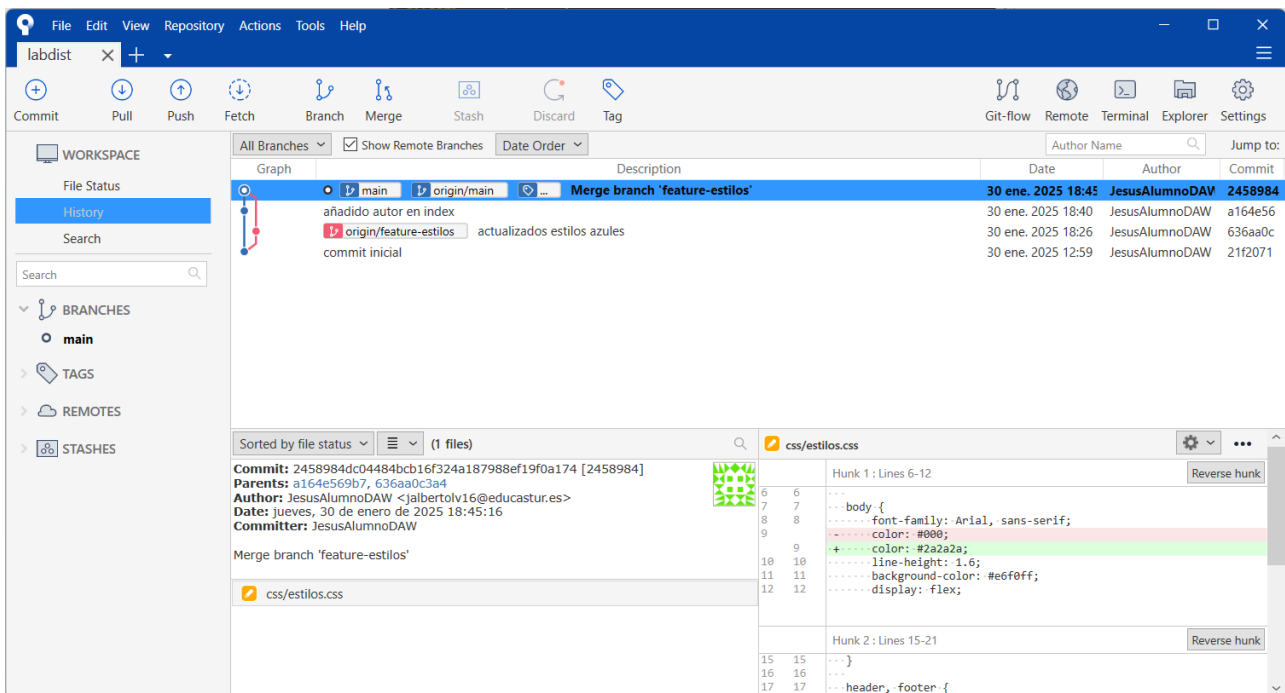
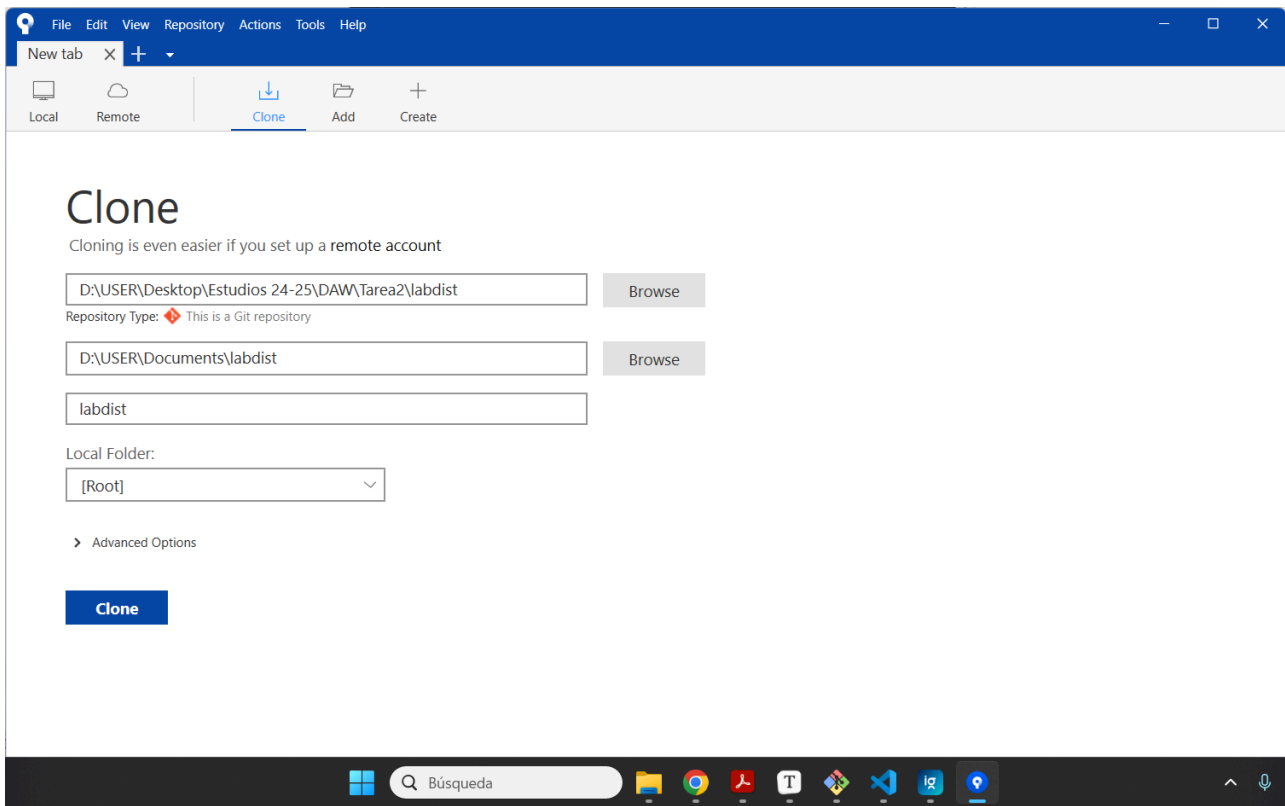
```
MINGW64:/d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git merge feature-estilos
hint: Waiting for your editor to close the file... unix2dos: converting file D:/
USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist/.git/MERGE_MSG to DOS format...
dos2unix: converting file D:/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist/.git
/MERGE_MSG to Unix format...
Merge made by the 'ort' strategy.
 css/estilos.css | 6 +++---
 1 file changed, 3 insertions(+), 3 deletions(-)

User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$ git log --oneline --graph --decorate --all
* 2458984 (HEAD -> main) Merge branch 'feature-estilos'
| \
| * 636aa0c (feature-estilos) actualizados estilos azules
* | a164e56 añadido autor en index
|/
* 21f2071 commit inicial

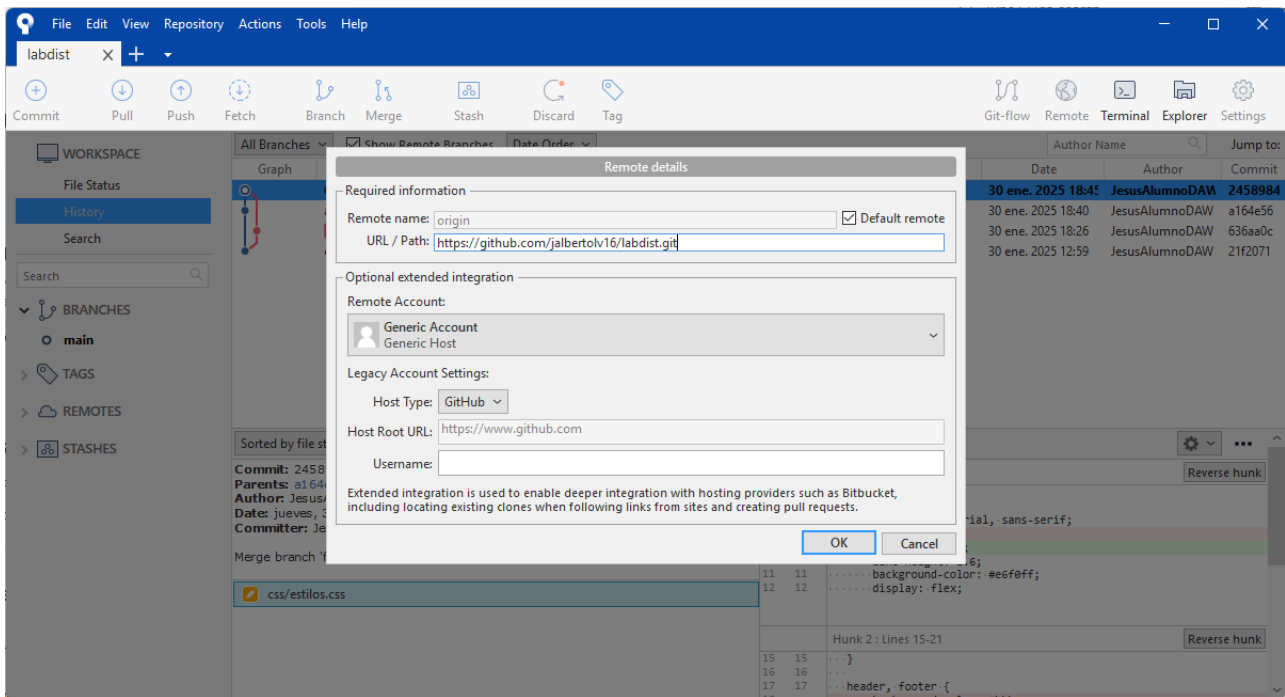
User@DESKTOP-DMGAIP0 MINGW64 /d/USER/Desktop/Estudios 24-25/DAW/Tarea2/labdist (
main)
$
```

TRABAJO EN REMOTO

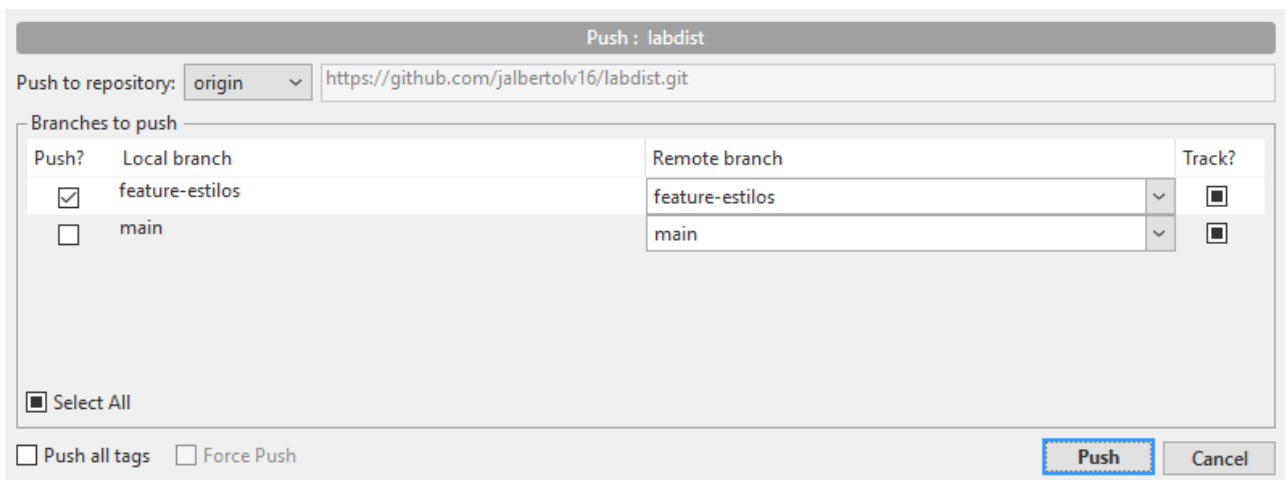
1-Continúa con el repositorio labdist . Añade el repositorio a Sourcetree.



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, algo similar a esto:



 Captura de pantalla 2025-01-30 202557



 Captura de pantalla 2025-01-30 202841

4-En el repositorio local, crea una rama feature-index . Añade el siguiente código dentro de la

. 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)

 Captura de pantalla 2025-01-31 185750

 Captura de pantalla 2025-01-31 190400

 Captura de pantalla 2025-02-01 191530

 Captura de pantalla 2025-02-01 191600

Los cambios ya están hechos en local, ahora vamos a ir a Push para subir los cambios a remoto.

 Captura de pantalla 2025-02-01 195354

 Captura de pantalla 2025-02-01 195757

Esto indica que se ha subido al remoto.

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

Comprobamos que estamos en la rama main, y en el Graphs que estamos parados en el punto donde queremos hacer el merge, con el boton derecho en ese punto

 Captura de pantalla 2025-02-01 195950

Seleccionamos merge

 Captura de pantalla 2025-02-01 200134

En esta imagen se puede comprobar que el merge fue realizado con éxito.

 Captura de pantalla 2025-02-01 202449

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

 Captura de pantalla 2025-02-01 203645

 Captura de pantalla 2025-02-01 203834

Aquí se puede comprobar el código original, lo que se ve en rojo es lo que se a eliminado y lo que se ve en verde lo que se ha añadido y en la imagen siguiente se puede comprobar que se ha hecho commit con los cambios.

 Captura de pantalla 2025-02-01 204411

7-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.)

Lo primero accederemos al último commit anterior al que queremos eliminar , clic derecho y nos dirá reset current branch this commit, al pinchar nos pedirá si deseamos conservar los cambios o eliminarlos, soft o hard, elegiremos hard.

 Captura de pantalla 2025-02-01 204631

 Captura de pantalla 2025-02-01 204718

 Captura de pantalla 2025-02-01 204913

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

 Captura de pantalla 2025-02-01 205515

Aquí se puede comprobar la nueva rama feature-mapa y el commit.

9-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 .

Rama subida al remoto en la siguiente imagen

 Captura de pantalla 2025-02-01 210059

En Github en la rama feature-mapa accedo al fichero contacto.html

 Captura de pantalla 2025-02-01 210240

Y el código modificado se puede ver

 Captura de pantalla 2025-02-01 210321

10-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

En Git pinchamos en Pull requests, previo revisar que estamos en la rama main ubicados

 Captura de pantalla 2025-02-01 210726

New pull request

 Captura de pantalla 2025-02-01 210925

Eliges la rama que deseas fusionar

 Captura de pantalla 2025-02-01 210938

Comprobamos que se va a fusionar tal cual deseamos la rama feature-mapa en main

 Captura de pantalla 2025-02-01 211116

Comprobamos los cambios sean los desados

 Captura de pantalla 2025-02-01 211133

Añadimos opcionalmente titulo y descripción

 Captura de pantalla 2025-02-01 211704

Comprobamos que no hay conflictos

 Captura de pantalla 2025-02-01 211731

 Captura de pantalla 2025-02-01 212150

 Captura de pantalla 2025-02-01 212235

 Captura de pantalla 2025-02-01 212327

 Captura de pantalla 2025-02-01 212356

Finalmente podemos comprobar que se ha realizado el merge correctamente y se han fusionado.

En sourcetree desde la rama main realizamos pull.

 Captura de pantalla 2025-02-01 214600

 Captura de pantalla 2025-02-01 214835

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".
(Fíjate en los números de línea de tu editor ...)

 Captura de pantalla 2025-02-02 112543

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".

 Captura de pantalla 2025-02-02 112925

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.

Nos colocamos en la rama main haciendo checkout y hacemos click derecho sobre la rama a fusionar, ya me indica una ventana emergente que hay conflictos así que sobre el archivo que tiene el conflicto hacemos click derecho y aparecerá esta imagen

 Captura de pantalla 2025-02-02 114852

Nos dará la opción de Resolve using "Mine", "Theirs ", Restart Merge... en mi caso elegí quedarme con mis cambios que como estoy ubicado en la rama main, son los cambios que se realizaron desde main.

 Captura de pantalla 2025-02-02 115430

 Captura de pantalla 2025-02-02 115457

 Captura de pantalla 2025-02-02 115741

En estas 3 últimas imágenes se puede comprobar como no existe ninguna notificación sobre el conflicto, ni en Sourcetree, ni en Visual-Estudio ni en GitHub.