

Ejercicio de Git - Proyecto labowebs

Proyecto realizado por Alejandro Luis Rey

Ejercicio de Git - Proyecto labowebs

- Trabajo en local
 - Cuestiones
- Trabajo en remoto
 - Cuestiones
- Conflictos
 - Cuestiones

Trabajo en local

Trabajo individual en tu ordenador - Esta parte se resuelve con comandos. Tu solución debe incluir el enunciado de cada cuestión, un bloque de código con los comandos GIT utilizados y captura/s de pantalla donde se vea la ejecución de los comandos y su salida.

Cuestiones

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.

```
cliente@cliente-22:/media/sf_labowebs$ git init
Inicializado repositorio Git vacio en /media/sf_labowebs/.git/
cliente@cliente-22:/media/sf_labowebs$ git add .
fatal: detected dubious ownership in repository at '/media/sf_labowebs'
To add an exception for this directory, call:

    git config --global --add safe.directory /media/sf_labowebs
cliente@cliente-22:/media/sf_labowebs$ git config --global --add safe.directory /media/sf_labowebs
cliente@cliente-22:/media/sf_labowebs$ git branch -M main
cliente@cliente-22:/media/sf_labowebs$ git commit -m "Master a Main"
En la rama main

Confirmación inicial

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que será confirmado)
    contacto.html
    css/
    index.html
    js/

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
cliente@cliente-22:/media/sf_labowebs$ git add .
cliente@cliente-22:/media/sf_labowebs$ git commit -m "Master a Main"
[main (commit-raiz) 0095ea3] Master a Main
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
cliente@cliente-22:/media/sf_labowebs$ git log --oneline
0095ea3 (HEAD -> main) Master a Main
cliente@cliente-22:/media/sf_labowebs$
```

```
git init
git config --global --add --safe.directory /media/sf_labowebs
git add .
git branch -M main
git commit -m "Master a Main"
git log --oneline
```

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.

```
cliente@cliente-22:/media/sf_labowebs$ echo "README.md" >> .gitignore
cliente@cliente-22:/media/sf_labowebs$ echo "LICENSE.txt" >> .gitignore
cliente@cliente-22:/media/sf_labowebs$ echo "passwords.txt" >> .gitignore
cliente@cliente-22:/media/sf_labowebs$ git add .gitignore
cliente@cliente-22:/media/sf_labowebs$ git commit -m ".gitignore actualizado"
[main 2d8faae] .gitignore actualizado
1 file changed, 3 insertions(+)
create mode 100644 .gitignore
cliente@cliente-22:/media/sf_labowebs$ git log --oneline
2d8faae (HEAD -> main) .gitignore actualizado
0095ea3 Master a Main
cliente@cliente-22:/media/sf_labowebs$
```

```
echo "README.md" >>.gitignore
echo "LICENSE.txt" >>.gitignore
echo "passwords.txt" >>.gitignore
git add .gitignore
git commit -m ".gitignore actualizado"
git log --oneline
```

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.

```

cliente@cliente-22:/media/sf_laboweb$ echo "Contenido del archivo README" > README.md
cliente@cliente-22:/media/sf_laboweb$ echo "Contenido del archivo LICENSE" > LICENSE.txt
cliente@cliente-22:/media/sf_laboweb$ echo "Contraseña privada: alexluis1969" > passwords.txt
cliente@cliente-22:/media/sf_laboweb$ git status
En la rama main
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
    modificados:    contacto.html
    modificados:    js/script.js

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
cliente@cliente-22:/media/sf_laboweb$ git check-ignore*
git: 'check-ignore*' no es un comando de git. Mira 'git --help'.

El comando más similar es
    check-ignore
cliente@cliente-22:/media/sf_laboweb$ git check-ignore *
LICENSE.txt
passwords.txt
README.md
cliente@cliente-22:/media/sf_laboweb$ git commit -m "Cambios extra"
En la rama main
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
    modificados:    contacto.html
    modificados:    js/script.js

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
cliente@cliente-22:/media/sf_laboweb$ git add .
cliente@cliente-22:/media/sf_laboweb$ git commit -m "Cambios extra"
[main 681f39a] Cambios extra
 2 files changed, 2 insertions(+), 6 deletions(-)
cliente@cliente-22:/media/sf_laboweb$

```

```

echo "Contenido del archivo README" > README.md
echo "Contenido del archivo LICENSE" > LICENSE.txt
echo "Contraseña privada: alexluis1969" > passwords.txt
git status
git check-ignore *
git add .
git commit -m "Cambios extra"

```

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

```

cliente@cliente-22:/media/sf_laboweb$ git checkout -b feature-estilos
Cambiado a nueva rama 'feature-estilos'
cliente@cliente-22:/media/sf_laboweb$

```

```
git checkout -b feature-estilos
```

- Modifica el archivo `estilos.css`:
 - propiedad color del `body` y de `h2`: `#2a2a2a`

```
body {
  font-family: Arial, sans-serif;
  color: #ff0000;
  line-height: 1.6;
  background-color: #e6f0ff;
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}

h2 {
  color: #ff0000;
  margin-bottom: 15px;
}
```

- propiedad `background-color` de `header` y `footer`: `#2a75ff`

```
header, footer {
  background-color: #008d00;
  color: #ffffff;
  text-align: center;
  padding: 1em 0;
}
```

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

```
cliente@cliente-22:/media/sf_labowebs$ git checkout -b feature-estilos
Cambiado a nueva rama 'feature-estilos'
cliente@cliente-22:/media/sf_labowebs$ git status
En la rama feature-estilos
Cambios no rastreados para el commit:
  (usa "git add/rm <archivo>..." para actualizar a lo que se le va a hacer commit)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
    modificados:  css/estilos.css
    borrados:     estilos.css

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
cliente@cliente-22:/media/sf_labowebs$ git add .
cliente@cliente-22:/media/sf_labowebs$ git status
En la rama feature-estilos
Cambios a ser confirmados:
  (usa "git restore --staged <archivo>..." para sacar del área de stage)
    modificados:  css/estilos.css
    borrados:     estilos.css

cliente@cliente-22:/media/sf_labowebs$ git commit -m "colores actualizados v1.2"
[feature-estilos 2c0ac11] colores actualizados v1.2
 2 files changed, 1 insertion(+), 10 deletions(-)
 delete mode 100644 estilos.css
```

```
git status
git add .
git status
git commit -m "colores actualizados v1.2"
```

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'

```
cliente@cliente-22:/media/sf_labowebs$ git checkout main
Cambiado a rama 'main'
```

```
git checkout main
```

```
<!--Autor Alejandro Luis Rey-->
```

```
<!-- Autor Alejandro Luis Rey -->
```

```
cliente@cliente-22:/media/sf_laboweb$ git log --oneline --graph --decorate
* 18b2286 (HEAD -> main) añadido autor al index
* a400a42 colores actualizados v1.3
* 9915015 actualizados los estilos
* 681f39a Cambios extra
* 2d8faae .gitignore actualizado
* 0095ea3 Master a Main
```

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

```
cliente@cliente-22:/media/sf_laboweb$ git add .
cliente@cliente-22:/media/sf_laboweb$ git commit -m "añadido autor al index"
[main 18b2286] añadido autor al index
1 file changed, 1 insertion(+)
```

```
git add .
git commit -m "añadido autor al index"
```

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

```
cliente@cliente-22:/media/sf_laboweb$ git commit -m "resolucion conflicto"
[main 448588b] resolucion conflicto
cliente@cliente-22:/media/sf_laboweb$ git merge feature-estilos
Ya está actualizado.
```

```
git commit -m "resolucion conflicto"
git merge feature-estilos
```

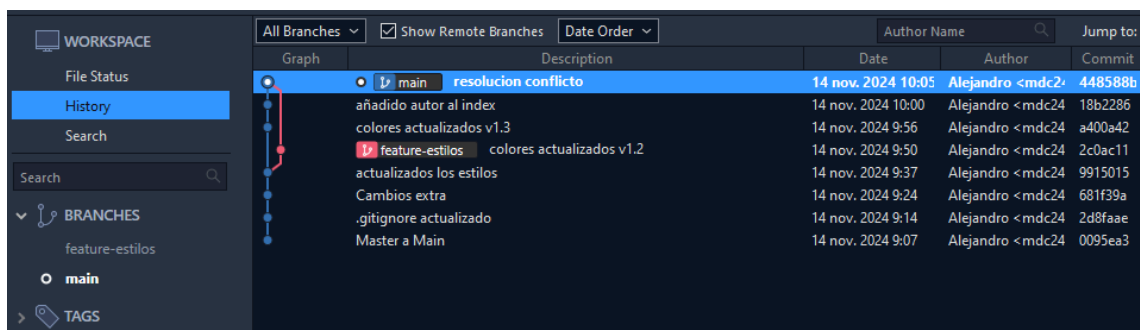
##

Trabajo en remoto

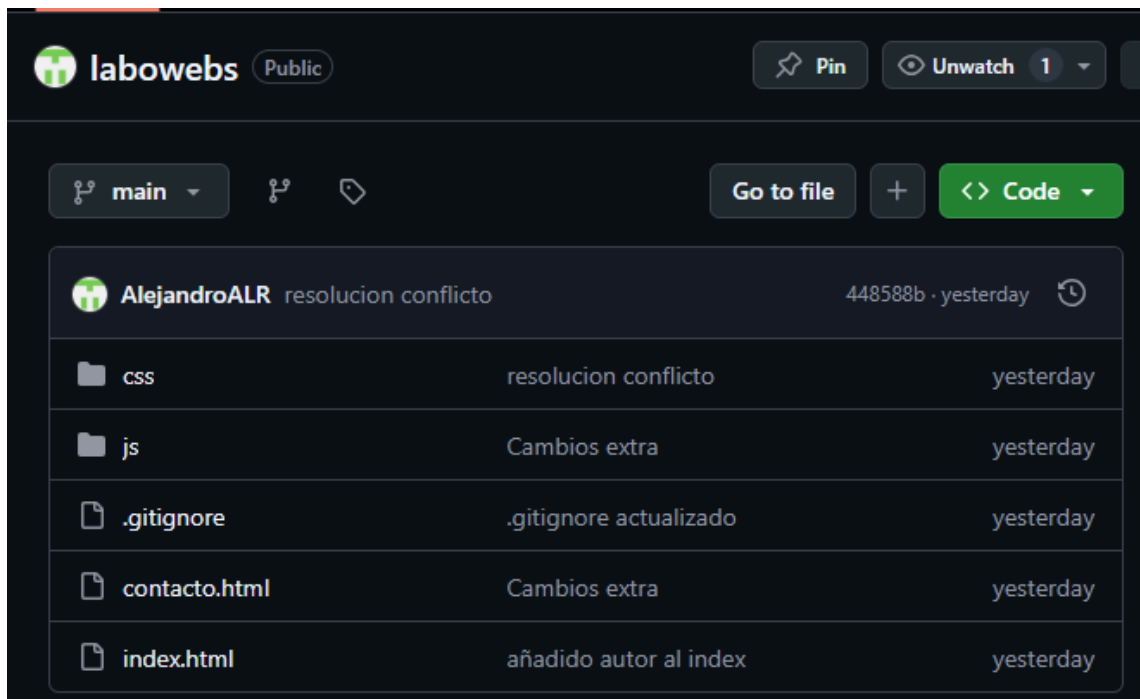
Esta parte se realizará con la herramienta gráfica `Sourcetree` y con `GitHub`. Para cada cuestión, incluye el enunciado y las capturas de pantalla donde se muestre la solución. Si es necesario, incluye alguna explicación

Cuestiones

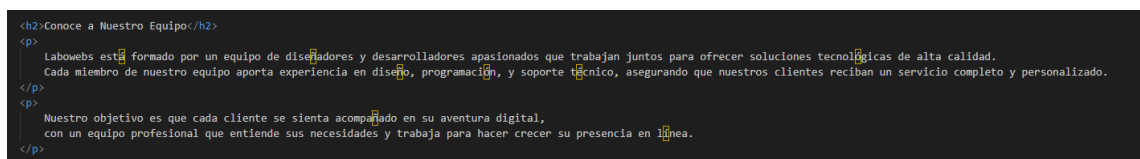
1. Continúa con el repositorio `Taboweb`. Añade el repositorio a Sourcetree.

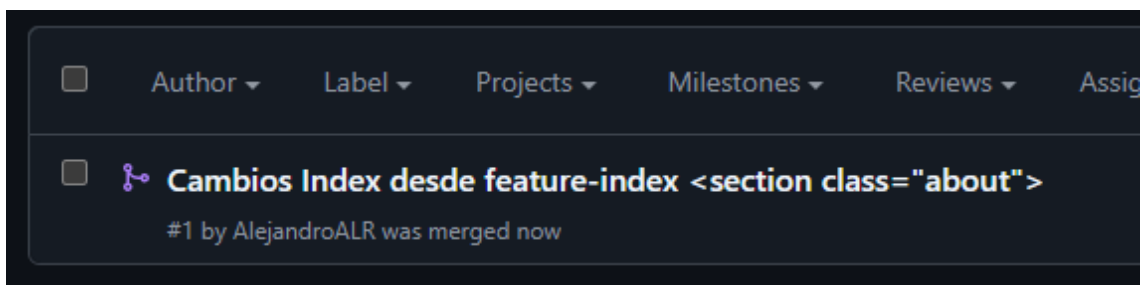


2. Crea un repositorio remoto y sube al remoto los ficheros de tu repositorio local. Debes subir todas las ramas.

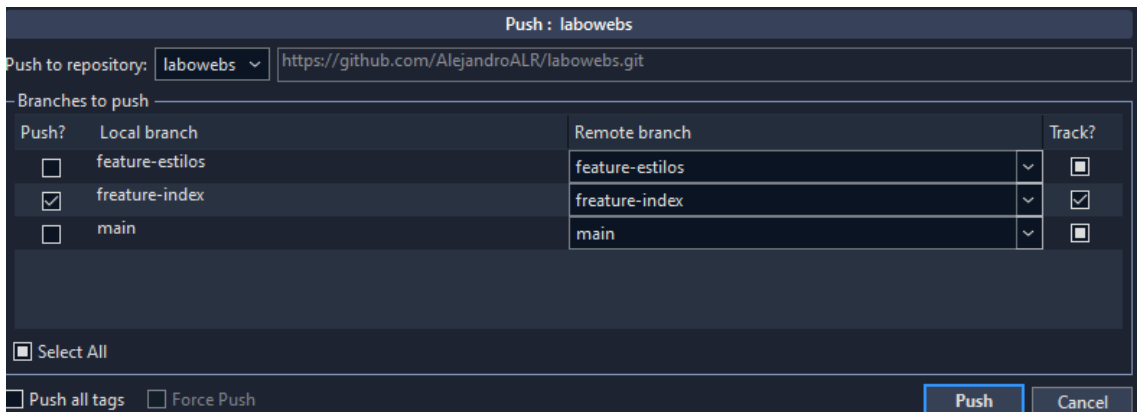


3. Crea una rama `feature-index`. Añade el siguiente código dentro de la `index.html`. Añade los cambios y crea un commit. Sube los cambios al remoto.

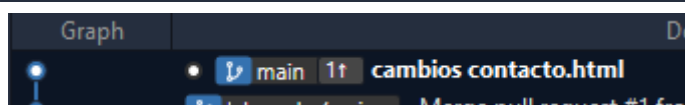
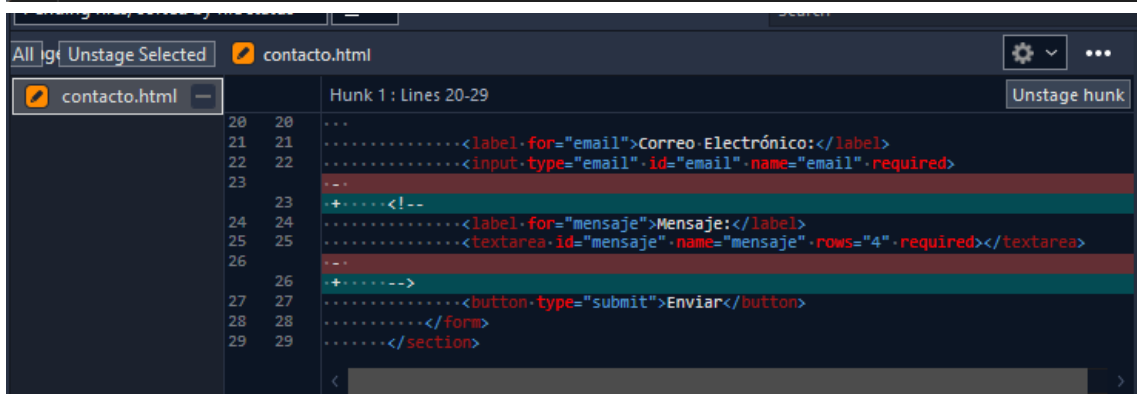




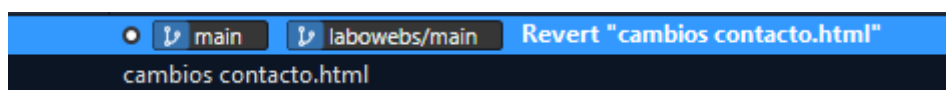
4. En el repositorio local, fusiona la rama `feature-index` en la rama `main`.



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 el commit anterior. Captura el estado actual del repositorio.



Antes

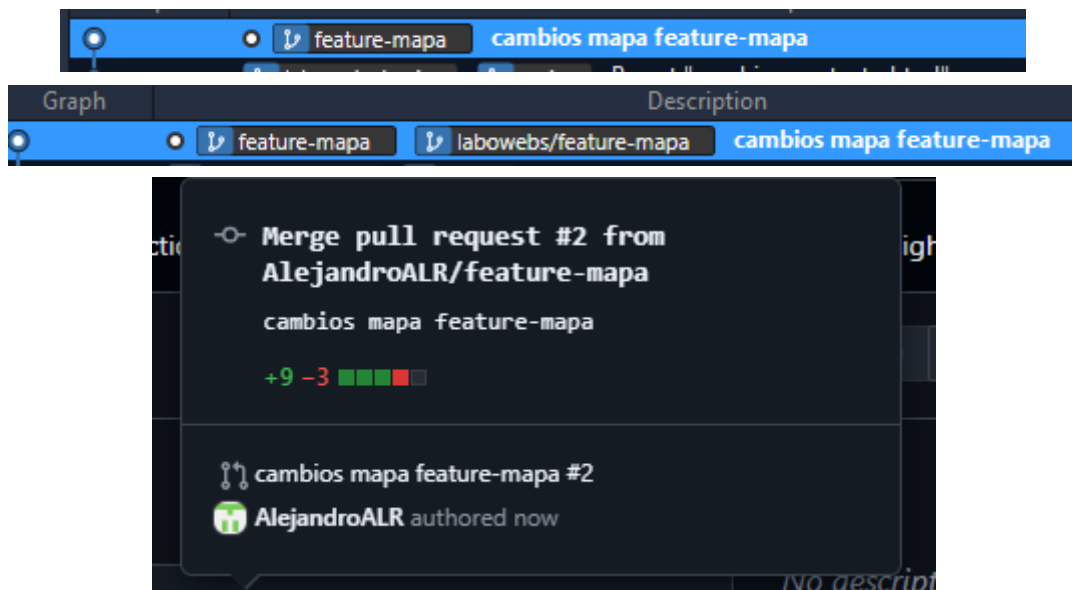
```
.....<input type="email" id="email" name="email" required>
+.....<!--
.....<label for="mensaje">Mensaje:</label>
.....<textarea id="mensaje" name="mensaje" rows="4" required></textarea>
+.....<!-->
.....<button type="submit">Enviar</button>
```

Después:

```
.....<input type="email" id="email" name="email" required>
+.....<!--
+.....<label for="mensaje">Mensaje:</label>
.....<textarea id="mensaje" name="mensaje" rows="4" required></textarea>
+.....<!-->
+.....<button type="submit">Enviar</button>
```

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`.

```
<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"
    width="600"
    height="450"
    style="border:0;"
    allowfullscreen=""
    loading="lazy">
  </iframe>
</section>
```



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

Merge pull request #2 from AlejandroALR/feature-mapa #3

Merged AlejandroALR merged 1 commit into `feature-mapa` from `main` now

Conversation **0**

Commits **1**

Checks **0**

Files changed **0**



AlejandroALR commented now

Owner ...

prueba pull request



Reviewers

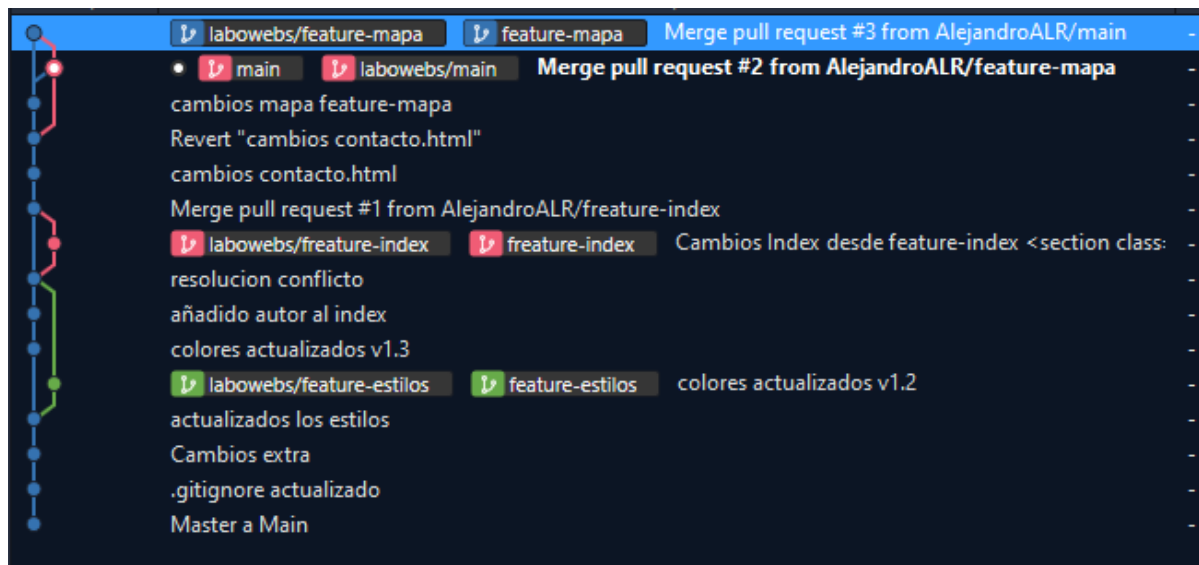
No reviews

Assignees

No one — [assign yourself](#)

Merge pull request #2 from AlejandroALR/feature-mapa **Verified** 6d57595

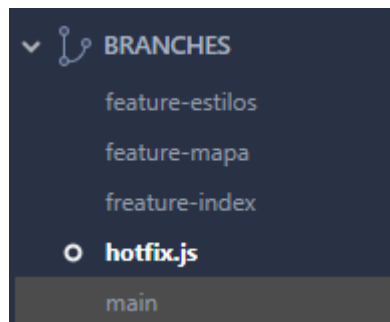
Labels



Conflictos

Cuestiones

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...)



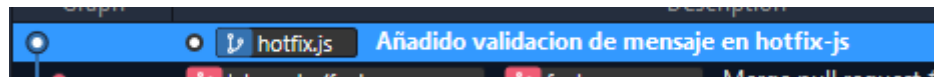
```

if (nombre.value.trim() === "") {
    alert("Por favor, ingrese su nombre.");
    valid = false;
}

if (!/^[\\w-\\.]+@([\\w-]+\\.){2,4}$/.test(email.value)) {
    alert("Por favor, ingrese un correo electrónico válido.");
    valid = false;
}

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

```



Push to repository: labowebs <https://github.com/AlejandroALR/labowebs.git>

Branches to push

Push?	Local branch	Remote branch	Track?
<input type="checkbox"/>	feature-estilos	feature-estilos	<input type="checkbox"/>
<input type="checkbox"/>	feature-mapa	feature-mapa	<input type="checkbox"/>
<input type="checkbox"/>	freature-index	freature-index	<input type="checkbox"/>
<input checked="" type="checkbox"/>	hotfix.js	hotfix.js	<input checked="" type="checkbox"/>
<input type="checkbox"/>	main	main	<input type="checkbox"/>

☐ Select All

☐ Push all tags ☐ Force Push

Push **Cancel**

- 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.

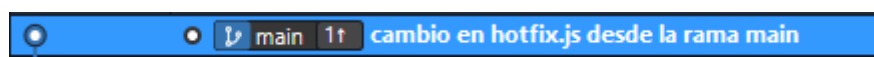
```

if (nombre.value.trim() === "") {
    alert("Por favor, ingrese su nombre.");
    valid = false;
}

if (!/^[\\w-\\.]+@([\\w-]+\\.){2,4}$/.test(email.value)) {
    alert("Por favor, ingrese un correo electrónico válido.");
    valid = false;
}

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

```



- 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

