

Ejercicio de Git - proyecto labowebs

Pelayo Rodríguez Álvarez

URL Repositorio GITHUB: <https://github.com/Pelayo87/labowebs.git>

[TOC]

<u>Trabajo en local</u>

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 commit -m "Primer commit o commit inicial"
$ git log
```

![image-20241114090256669](./EJERCICIO GIT- labowebs.assets/image-20241114090256669.png)

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.**

```
$ echo "README.md" >> .gitignore
$ echo "LICENSE.txt" >> .gitignore
$ echo "passwords.txt" >> .gitignore
$ git add .gitignore
$ git commit -m "Añadido archivo .gitignore que ignora README.md, LICENSE.txt y passwords.txt"
$ git log --oneline
```

![image-20241114090857878](./EJERCICIO GIT- labowebs.assets/image-20241114090857878.png)

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 "Leeme" > README.md
$ echo "Licencia" > LICENSE.txt
$ echo "Contraseña: labowebs" > passwords.txt
$ git status
$ git status --ignored
```

![image-20241114091314900](./EJERCICIO GIT- labowebs.assets/image-20241114091314900.png)

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
$ notepad estilos.css
$ git status
$ git add estilos.css
$ git commit -m "actualizados estilos con nuevos colores"
```

![image-20241114092659342](./EJERCICIO GIT- labowebs.assets/image-20241114092659342.png)

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
$ notepad index.html
$ git status
$ git add index.html
$ git commit -m "añadido autor en index"
$ git log --oneline --graph --decorate
```

![image-20241114093318732](./EJERCICIO GIT- labowebs.assets/image-20241114093318732.png)

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

![image-20241114093706605](./EJERCICIO GIT- labowebs.assets/image-20241114093706605.png)

Trabajo en remoto

7. **Continúa con el repositorio labowebs . Añade el repositorio a Sourcetree.**

![image-20241114094433359](./EJERCICIO GIT- labowebs.assets/image-20241114094433359.png)

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

```
$ git remote add origin https://github.com/Pelayo87/labowebs.git
$ git push --all origin
$ git push --tags
```

![image-20241114094947793](./EJERCICIO GIT- labowebs.assets/image-20241114094947793.png)

![image-20241114094823891](./EJERCICIO GIT- labowebs.assets/image-20241114094823891.png)

9. **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.**

```
$ git checkout -b feature-index
$ code index.html
$ git add index.html
$ git commit -m "Añadido contenido en section.about en index.html"
$ git push origin feature-index
```

![image-20241114095530805](./EJERCICIO GIT- laboweb.assets/image-20241114095530805.png)

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

```
$ git checkout main
$ git merge feature-index
$ git status
$ git push origin main
```

![image-20241114095758345](./EJERCICIO GIT- laboweb.assets/image-20241114095758345.png)

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.**

```
$ code contacto.html
$ git status
$ git diff contacto.html
$ git add contacto.html
$ git commit -m "Eliminadas líneas innecesarias de contacto.html"
```

![image-20241114100215401](./EJERCICIO GIT- laboweb.assets/image-20241114100215401.png)

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

```
$ git reset --hard HEAD~1
$ git status
```

![image-20241114100354696](./EJERCICIO GIT- laboweb.assets/image-20241114100354696.png)

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 .**

```
$ git checkout -b feature-mapa
$ code contacto.html
$ git add contacto.html
$ git commit -m "Añadido mapa de ubicación en contacto.html"
$ git push origin feature-mapa
```

![image-20241114100808381](./EJERCICIO GIT- laboweb.assets/image-20241114100808381.png)

![image-20241114100921071](./EJERCICIO GIT- laboweb.assets/image-20241114100921071.png)

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

![image-20241114101233883](./EJERCICIO GIT- labowebs.assets/image-20241114101233883.png)

![image-20241114101323304](./EJERCICIO GIT- labowebs.assets/image-20241114101323304.png)

<u>Conflictos</u>

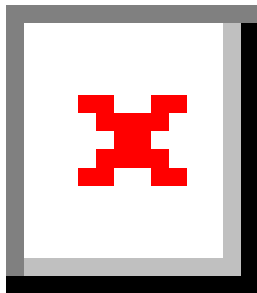
9. Crea una rama hotfix-js . Cámbiate a ella. Añade este código en el fichero script.js . Confirma el cambio y haz un commit. (Fíjate en los números de línea...)

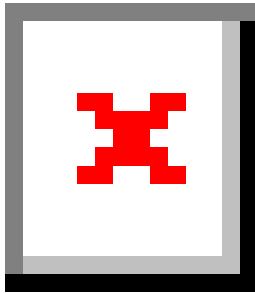
```
$ git checkout -b hotfix-js
$ code script.js
$ git add script.js
$ git commit -m "Hotfix aplicado en script.js para corregir el error"
$ git push origin hotfix-js
$ git push origin hotfix-js
```

![image-20241114101708278](./EJERCICIO GIT- labowebs.assets/image-20241114101708278.png)

10. 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.

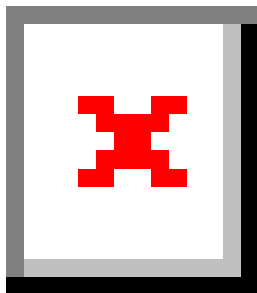
```
$ git checkout main
$ code js/script.js
$ git add script.js
$ git commit -m "Añadido validación de mensaje vacío en script.js en la rama main"
```





11. **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.**

```
$ git merge hotfix-js
//RESUELVO EL CONFLICTO
$ code js/script.js
$ git add script.js
$ git commit -m "Resuelto conflicto entre hotfix-js y main en script.js"
$ git push origin main
```



Subida de documentación

En la carpeta del proyecto labowebs añade una carpeta docs . Copia en esa carpeta el fichero markdown y la carpeta con las imágenes. Incluye también el pdf . Añade todo al repositorio. Súbelo al remoto.

```
$ mkdir docs  
$ git add docs/  
$ git commit -m "Añadida carpeta docs con archivos Markdown, imágenes y PDF"  
$ git push origin main
```

