

Ejercicios git

En los siguientes ejercicios vamos a trabajar con una página web que es un portfolio, para poder ir visualizando el contenido simplemente tienes que abrir index.html en el navegador.

Ejercicio 1 - Prepare basic git environment

Vamos a preparar el proyecto 'portfolio'

a) Crea un repositorio nuevo local llamado portfolio

mkdir portfolio

cd portfolio

git init

b) Extrae los ficheros que te ha proporcionado el profesor en dicho repositorio.

Extraer.

c) Crea un repositorio privado nuevo en github llamado 'portfolio' da acceso al usuario del profesor e indica aquí la url

<https://github.com/sandra86code/portfolio.git>

d) Enlaza tu repositorio local con el repositorio remoto creado en el apartado anterior

git remote add origin <https://github.com/sandra86code/portfolio.git>

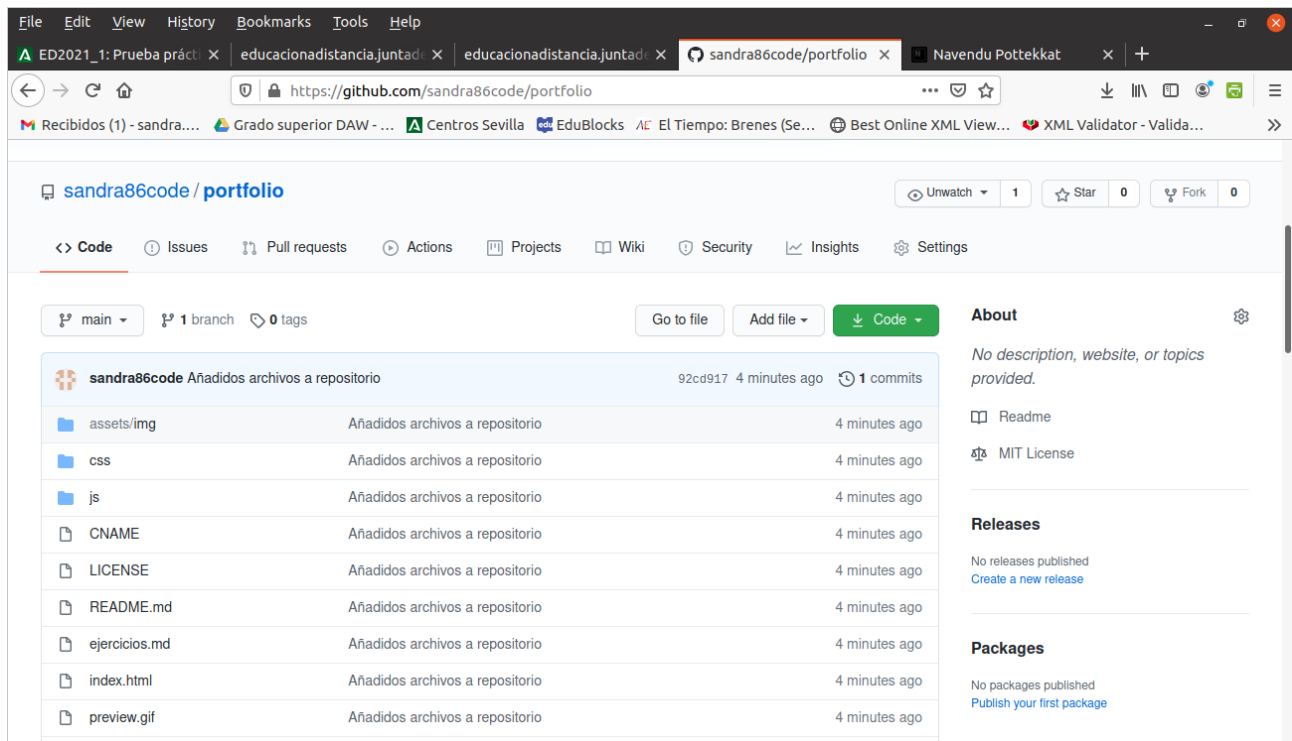
e) Haz lo necesario para que todos los ficheros estén disponibles en el repositorio remoto.

Git add .

Git commit -m "Añadidos archivos a repositorio".

Git branch -M main

git push -u origin main



f) Crea un directorio en tu carpeta personal llamado proyectosGit. ¡¡¡OJO NO LO HAGAS DENTRO DEL REPOSITORIO ANTERIOR!!!

cd ..

mkdir proyectosGit

cd proyectosGit

Ejercicio 2

En este ejercicio vamos crear dos copias del repositorio de github en nuestro directorio de proyectosGit para simular que trabajamos en dos ordenadores distintos uno será el de Julia y otro el de Pablo.

a) Crea dos copias del repositorio portfolio de github llamadas portJulia y portPablo.

git clone <https://github.com/sandra86code/portfolio.git> portJulia

git clone <https://github.com/sandra86code/portfolio.git> portPablo

Ejercicio 3

Cámbiate al repositorio de portJulia y realiza los siguientes ejercicios en él.

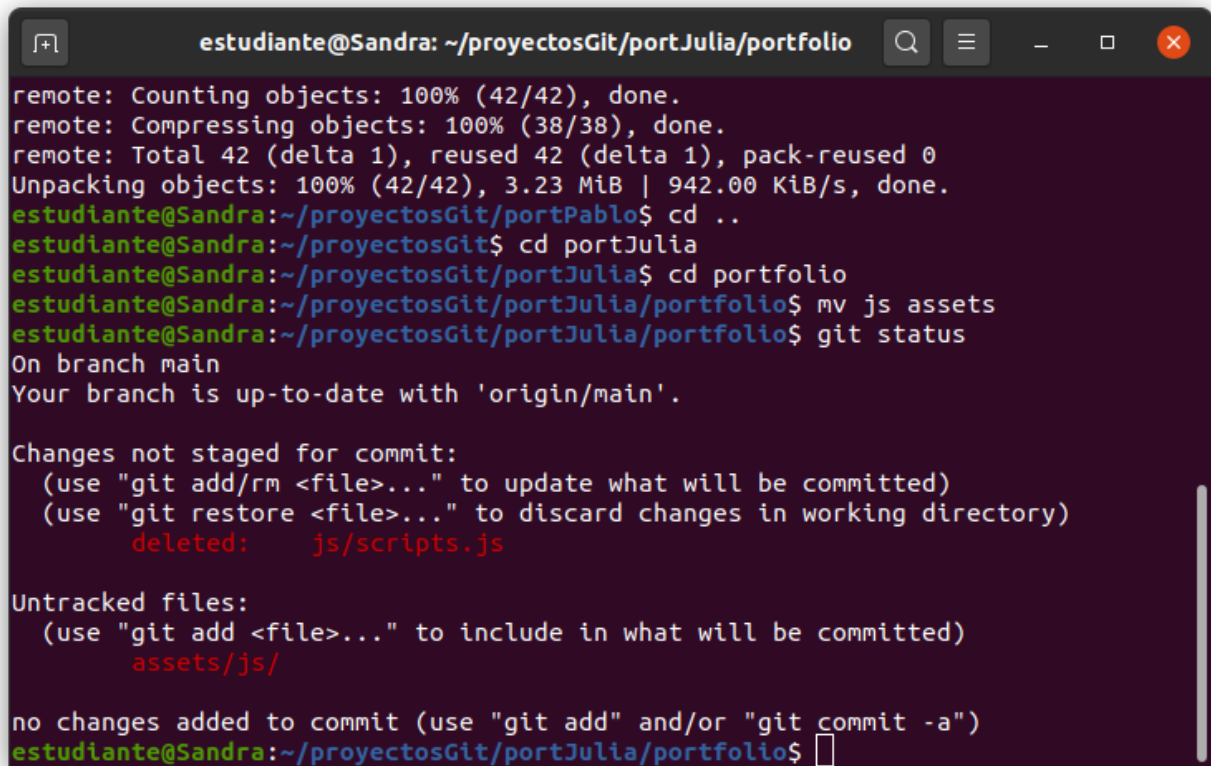
cd portJulia

a) Vamos a organizar las carpetas:

- Mueve el directorio js al directorio assets con el comando mv

mv js/ assets/

- Comprueba el estado del repositorio y muéstralo aquí



```
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 42 (delta 1), reused 42 (delta 1), pack-reused 0
Unpacking objects: 100% (42/42), 3.23 MiB | 942.00 KiB/s, done.
estudiante@Sandra:~/proyectosGit/portPablo$ cd ..
estudiante@Sandra:~/proyectosGit$ cd portJulia
estudiante@Sandra:~/proyectosGit/portJulia$ cd portfolio
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ mv js assets
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ git status
On branch main
Your branch is up-to-date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    js/scripts.js

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

no changes added to commit (use "git add" and/or "git commit -a")
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$
```

- Mueve el directorio css al directorio assets con el comando git mv

git mv css/ assets/

- Comprueba el estado del repositorio y muéstralo aquí

```
estudiante@Sandra: ~/proyectosGit/portJulia/portfolio
tJulia/portfolio
estudiante@Sandra:~/proyectosGit/portJulia/portfolio/assets$ cd ..
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ git mv css assets
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ git status
On branch main
Your branch is up-to-date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    css/body.css -> assets/css/body.css
    renamed:    css/heading.css -> assets/css/heading.css
    renamed:    css/styles.css -> assets/css/styles.css

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    js/scripts.js

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

estudiante@Sandra:~/proyectosGit/portJulia/portfolio$
```

b) Indica las diferencias que encuentras entre mover los ficheros de una forma o de otra.

Con `git mv css assets` los archivos del directorio `css` son renombrados (por su ruta) y están en el Staging area, por lo que es necesario hacer un `git commit` para llevarlos al directorio local. Con `mv css assets`, el archivo del directorio `js` está en el Working area, no en el staging area.

c) Añade los cambios al área de preparación.

Git add .

d) Comprueba el estado del repositorio y muéstralo aquí

```
estudiante@Sandra: ~/proyectosGit/portJulia/portfolio

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    js/scripts.js

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

estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ git add .
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ git status
On branch main
Your branch is up-to-date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    css/body.css -> assets/css/body.css
        renamed:    css/heading.css -> assets/css/heading.css
        renamed:    css/styles.css -> assets/css/styles.css
        renamed:    js/scripts.js -> assets/js/scripts.js

estudiante@Sandra:~/proyectosGit/portJulia/portfolio$
```

e) Guarda los cambios con un mensaje acorde a los cambios realizados.

git commit -m "Movidos directorios css y js a directorio assets"

f) Olvidaste cambiar el código para que funciones correctamente la página, para ello tienes que modificar en el fichero index.html las siguientes líneas:

- 29, 30 y 31 añade assets/ delante de css. Las líneas quedarían así:

```
<link href="assets/css/styles.css" rel="stylesheet">
<link rel="stylesheet" href="assets/css/heading.css">
<link rel="stylesheet" href="assets/css/body.css">
```

- 394 añade assets/ delante de js

nano index.html

CtrlW buscar

CtrlR reemplazar

g) Añade los cambios al área de preparación y añade los cambios al commit anterior modificando el mensaje para que refleje también los nuevos cambios.

Git add index.html

git commit --amend

h) Muestra los commits realizados en formato de una línea y pega aquí el resultado.

git log --oneline

```
estudiante@Sandra: ~/proyectosGit/portJulia/portfolio
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:    css/body.css -> assets/css/body.css
    renamed:    css/heading.css -> assets/css/heading.css
    renamed:    css/styles.css -> assets/css/styles.css
    renamed:    js/scripts.js -> assets/js/scripts.js
    modified:    index.html

estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ git commit -m "Movido
s css y jss a assets y modificado index.html"
[main da5c2f0] Movidos css y jss a assets y modificado index.html
 5 files changed, 5 insertions(+), 4 deletions(-)
 rename {css => assets/css}/body.css (100%)
 rename {css => assets/css}/heading.css (100%)
 rename {css => assets/css}/styles.css (100%)
 rename {js => assets/js}/scripts.js (100%)
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$ git log --oneline
da5c2f0 (HEAD -> main) Movidos css y jss a assets y modificado index.html
92cd917 (origin/main, origin/HEAD) Añadidos archivos a repositorio
estudiante@Sandra:~/proyectosGit/portJulia/portfolio$
```

i) Envía los cambios al repositorio remoto
git push

j) Modifica el fichero index.html, cambia en la línea 58 avataaars.svg por computer.png
nano index.html
CtrlW buscar
CtrlR reemplazar

k) Comprueba el estado del repositorio y las diferencias del fichero.
Git status
Falta captura de pantalla
git diff

l) Añade el fichero al área de preparación.
Git add index.html

m) Vuelve a mostrar los cambios en el fichero.
git diff --staged

n) Envía los cambios al repositorio remoto.
git commit -m "Modificada línea 58 index.html"
git push

Ejercicio 4

Vamos a trabajar en el repositorio de Pablo, en portPablo.

Cd ..

cd portPablo

a) Actualiza el repositorio portPablo y analiza los mensajes

git pull

5 archivos han sido cambiados, ha habido 6 inserciones y 5 eliminaciones

Me hace una lista de los archivos que han sido renombrados.

b) Crea un nuevo directorio llamado 'docs'

mkdir docs

- Comprueba el estado del repositorio. ¿Qué ha pasado?

Me dice que no hay cambios pendientes de commit porque he creado una carpeta vacía y Git no hace seguimiento de carpetas vacías.

- Haz lo necesario para que los cambios aparezcan en el repositorio remoto

cd docs

touch .keep

git add .

git commit -m "Añadido directorio docs"

cd ..

git push

c) Edita el fichero html y cambia todas las ocurrencias de 'Navendu Pottekkat' por tu nombre.

nano index.html

CtrlW buscar

CtrlR reemplazar

- Comprueba los cambios al repositorio remoto e indica lo que observas.

Que sin hacer git add index.html y git commit -m "xxxx" no se pueden subir los cambios que he hecho en un archivo porque no están guardados en el repositorio local.

d) Envía los cambios al repositorio remoto

git add index.html

git commit -m "Modificado index.html con mi nombre"

git push

Ejercicio 5

Volvemos a trabajar en el repositorio de Julia, en portJulia.

cd ..

cd portJulia

a) Crea el fichero page2.html con el siguiente contenido:

```
<!doctype html>
  <html lang="en">
    <head>
      <!-- Required meta tags -->
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

      <!-- Bootstrap CSS -->
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg
FAW/dAiS6JXm" crossorigin="anonymous">

      <title>Hello, world!</title>
    </head>
    <body>
      <h1>Hello, world!</h1>

      <!-- Optional JavaScript -->
      <!-- jQuery first, then Popper.js, then Bootstrap JS -->
      <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtlkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG
FF93hXpG5KkN" crossorigin="anonymous"></script>
      <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk
vXusvfa0b4Q" crossorigin="anonymous"></script>
      <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5
+76PVCmYI" crossorigin="anonymous"></script>
    </body>
  </html>
```

nano page2.html

b) Haz lo necesario para enviar los cambios al repositorio

git add page2.html

git commit -m "Añadido page2.html"

git pull

git push

Ejercicio 6

Volvemos a trabajar en portPablo:

cd ..

cd portPablo

a) Elimina el fichero LICENSE, utiliza únicamente 'rm'.

rm LICENSE

- comprueba el estado del repositorio analizando los mensajes.

En el Working area tengo el archivo eliminado LICENSE, pero no en el area de preparacion.

- Deshaz el cambio.

Git restore LICENSE

b) Elimina el fichero LICENSE de nuevo, pero esta vez utiliza 'git rm'

git rm LICENSE

- Comprueba el estado del repositorio analizando los mensajes.

Git status

En este caso, el archivo eliminado está en el staging area.

- Deshaz el cambio.

Git restore --stage LICENSE

Git restore LICENSE

c) Elimina una vez más LICENSE y haz commit con los cambios.

git rm LICENSE

git commit -m "Eliminado LICENSE"

d) Recupera el fichero.

Git reset --hard HEAD~1

d) Por último elimina el fichero del repositorio y haz que los cambios se reflejen en el repositorio remoto.

git rm LICENSE

git commit -m "Eliminado archivo LICENSE"

git pull

git push

Ejercicio 7

En portJulia:

a) Actualiza el repositorio.

```
cd ../portJulia  
git pull
```

b) Cambia en la línea 68 Machine Learning - Computer Vision por Frontend Developer y ML on Edge por Estudiante en IES Jacarandá

```
nano index.html
```

c) Envía los cambios al repositorio remoto.

```
git commit -am "Modificado index.html"  
git push
```

En portPablo:

```
cd ../portPablo
```

a) NO ACTUALICES EL REPOSITORIO

b) Camba en la línea 68 "Machine Learning - Computer Vision por Fullstack Developer y ML on Edge por Estudiante DAW en IES Jacarandá

```
nano index.html
```

c) Realiza los pasos para conseguir que en el repositorio remoto quede como lo acabas de modificar.

```
git commit -am "Editado perfil"  
git pull  
git push
```

hay un conflicto y arreglamos el conflicto, dejándolo como la versión HEAD.

```
git commit -am "Resuelto conflicto perfil"  
git push
```

d) Averigua quién y cuando se creó el fichero page2.html

```
git blame page2.html o git log page2.html (sirven los dos porque no pregunta quién editó qué línea, para lo que solo valdría git blame)
```

Ejercicio 8

En portJulia:

cd ../portJulia

a) Actualiza el repositorio.

git pull

b) Crea una rama llamada cambiandoProyectos.

git branch cambiandoProyectos

c) Cámbiate a dicha rama.

git checkout cambiandoProyectos

d) Modifica el nombre de algún proyecto en el fichero index.html (Si no sabes como pregunta al profesor) y envía los cambios al repositorio remoto a su rama.

Search Awesome README y reemplazar por Proyecto superchuli

commit -am "Modificado proyecto"

git push origin cambiandoProyectos

e) Cámbiate a la rama main.

git checkout main

f) Añade cualquier contenido al fichero page2.html.

nano page2.html

git commit -am "Modificada page2"

g) Incorpora los cambios de la rama master a la rama cambiandoProyectos.

git push

git checkout cambiandoProyectos

git merge main

h) Muestra los cambios que se han ido produciendo en el repositorio de manera que se muestren las ramas gráficamente.

git log --oneline --graph

i) Añade algún contenido al fichero page2.html y sube los cambios al repositorio.

nano page2.html

git commit -am "Modificada page2 again"

j) Incorpora los cambios de la rama cambiandoProyectos a la rama main.

git push origin cambiandoProyectos

git checkout main

git merge cambiandoPoryectos

k) Elimina la rama cambiandoProyectos.

git branch -d cambiandoProyectos

l) Sube los cambios al repositorio remoto.

git push