28-1-2023

Comandos Git y Github.

Práctica 1 – Programación de Aplicaciones Telemáticas

Juan Orlando Cives Espejo

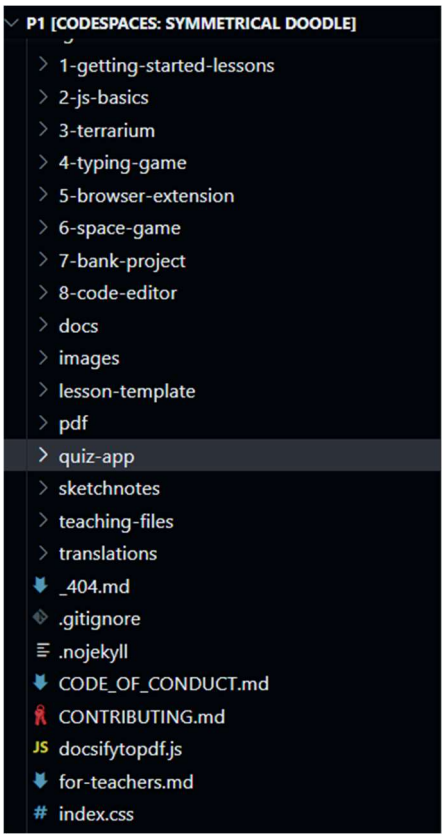
1. Comando git clone.

En primer lugar, se busca en *Github* un repositorio interesante para crear un ***fork*** de él y poder usar los distintos comandos sobre él. Esto se debe a que en clase ya se había realizado previamente un *fork* de la página https://github.com/gitt-3-pat/p1, y sobre él ya se habían ejecutado los comandos Git que se usarán también en esta práctica. El único propósito de buscar este nuevo repositorio es ejecutar estos comandos en un entorno distinto al empleado en clase.

Tras navegar, se encuentra el siguiente repositorio: https://github.com/microsoft/Web-Dev-For-Beginners. Este repositorio de Microsoft es una introducción al desarrollo Web. Seguidamente, hago uso del siguiente comando:

```
git clone https://github.com/JuanCivesEspejo/Web-Dev-For-Beginners
```

Este primer comando ***git clone*** permite copiar el repositorio completo en mi máquina local. Además, al hacer uso de este código en el ***Codespace*** del repositorio ***p1*** en el que se realiza la práctica, también muestra en este mismo repositorio todo el Directorio de ficheros del repositorio de Microsoft tal y como se muestra en la siguiente imagen:



Asimismo, el resultado que se mostró por pantalla al ejecutar el comando fue el siguiente:

```
Cloning into 'Web-Dev-For-Beginners'...
remote: Enumerating objects: 13419, done.
remote: Counting objects: 100% (2757/2757), done.
remote: Compressing objects: 100% (363/363), done.
```

```
remote: Total 13419 (delta 2500), reused 2484 (delta 2391), pack-reused 10662
Receiving objects: 100% (13419/13419), 85.07 MiB | 22.88 MiB/s, done.
Resolving deltas: 100% (9094/9094), done.
```

2. Comando git status.

A continuación, se pretende realizar un cambio en la rama *main* del repositorio. Los cambios se realizan en ramas independientes y separadas de la rama principal, y se efectúan en tres pasos: ***add, commit*** y ***push***. Previamente, se hace uso del comando ***git status***:

```
git status
```

Este comando muestra dos cosas: el estado actual del repositorio, indicando qué archivos se han eliminado, se han modificado o se han añadido, y la rama en la que te encuentras, así como los cambios que se han realizado en ella. Al usar este comando, la salida ha sido la siguiente:

```
Your branch is up to date with 'origin/main'.

Untracked files:
 (use "git add <file>..." to include in what will be committed)
    Web-Dev-For-Beginners/
```

La salida indica que, estando en la rama principal, se ha detectado que se han añadido nuevos repositorios, en este caso, el repositorio “Web-Dev-For-Beginners”.

3. Comandos git add, commit y push

Para añadir este nuevo directorio de ficheros a mi repositorio ***p1*** deberé en primer lugar hacer uso de este comando:

```
git add .
```

Este comando agrega todos los archivos que hayan sido modificados con un mismo propósito y los prepara para formar parte del siguiente ***commit***. No obstante, la salida que muestra este comando es interesante:

```
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> Web-Dev-For-Beginners
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached Web-Dev-For-Beginners
hint:
hint: See "git help submodule" for more information.
```

Por ello, se investiga la utilidad del comando **git submodule.** Este comando, a diferencia del comando ***git add*** se utiliza para agregar un submódulo a tu proyecto, siendo este submódulo un proyecto independiente con su propio repositorio, historial, ramas… Por lo tanto, se emplea este comando, en lugar de usar el comando ***git add***. Previamente, se elimina el *path* de Web-Dev-For-Beginners, añadido mediante el comando *git add,* para añadirlo como submódulo con el comando *git submodule*. Para ello, se emplea el siguiente comando:

```
git rm --cached Web-Dev-For-Beginners
```

Con este comando se elimina el repositorio que he intentado agregar con el comando ***git add.*** Para que tenga lugar su eliminación, también será necesario hacer uso de los comandos ***git commit*** y ***git push***.

Tras eliminar satisfactoriamente este repositorio, se ejecuta el comando sugerido en el aviso anterior:

```
git submodule add https://github.com/microsoft/Web-Dev-For-Beginners
Web-Dev-For-Beginners
```

Seguidamente, se hace uso del comando ***git commit*** de la siguiente forma:

```
git commit -m "Adding Submodule"
```

El comando ***git commit*** integra un conjunto de cambios en un proyecto que ya han sido verificados, pero que Git no cambiará hasta que no se le indique con el correspondiente ***git push***. El comando *-m* se utiliza para introducir un mensaje que resuma los cambios que se integran en ese ***commit***, indicando en este caso la incorporación de un submódulo al proyecto principal. La salida obtenida es a siguiente:

```
[main 1f0a3c1] Adding Submodule
 2 files changed, 7 insertions(+)
 create mode 100644 .gitmodules
 create mode 160000 Web-Dev-For-Beginners
```

Finalmente, se hace uso del comando ***git push*** de esta forma:

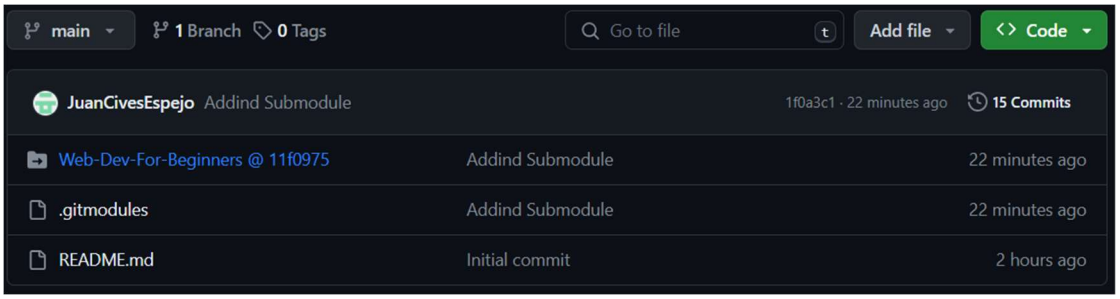
```
git push origin
```

Este comando permite subir los ***commits***, es decir, los cambios que han sido previamente verificados, al repositorio ***origin*** en la rama principal. La salida obtenida de este comando es la siguiente:

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 457 bytes | 457.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JuanCivesEspejo/p1
```

```
   8c1f35d..1f0a3c1  main -> main
```

El repositorio de la práctica 1 queda de la siguiente forma:



4. Comando git checkout

Asimismo, el comando **git checkout -b** permite crear y cambiar a una nueva rama para realizar modificaciones sin afectar al *main*. El comando ejecutado ha sido el siguiente:

```
git checkout -b feat/Add-Tittle-ReadMe
```

La nueva rama será por tanto ***feat/Add-Tittle-ReadMe*** y como su propio nombre indica, permitirá realizar modificaciones en el título del fichero ReadMe.md. La salida del comando es la siguiente:

```
Switched to a new branch 'feat/Add-Tittle-ReadMe'
```

A continuación, se cambia el título del fichero ReadMe.md, y en lugar de ser **"p1",** pasará a llamarse **"Práctica 1 – Entorno de desarrollo".** Tras realizar los cambios pertinentes en el título del ReadMe.md, será necesario ejecutar la siguiente secuencia de comandos:

```
git add .
git commit -m "feat: Change ReadMe title"
git push origin feat/Add-Tittle-ReadMe
```

Los cuales, tal y como se indicó previamente, permiten realizar cambios en el repositorio. Finalmente, se mezclan ambas ramas, para unir los cambios realizados y se hace uso del comando ***git checkout main*** para regresar a la rama ***main***. El resultado del cambio del título es el siguiente: