

Práctica .GIT

DM1E

Adrián Caballero Orasio | Entornos de desarrollo | 19 de enero de 2020

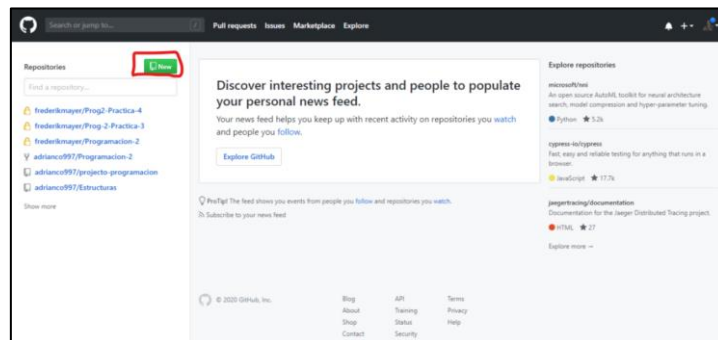
Contenido

Inicialización: https://github.com/adrianco997/FP_ESPECIALIDAD_1920	3
Eclipse	4
GIT Bash	5

Inicialización:

https://github.com/adrianco997/FP_ESPECIALIDAD_1920

Crear una cuenta / entrar en tu cuenta personal y seleccionar crear repositorio:



Escribir los datos necesarios y crear el repositorio:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: Repository name:

Great repository names are: [Your new repository will be created as FP_ESPECIALIDAD_1920- >barnacle?](#)

Description (optional):

☒ Public: Anyone can see this repository. You choose who can commit.

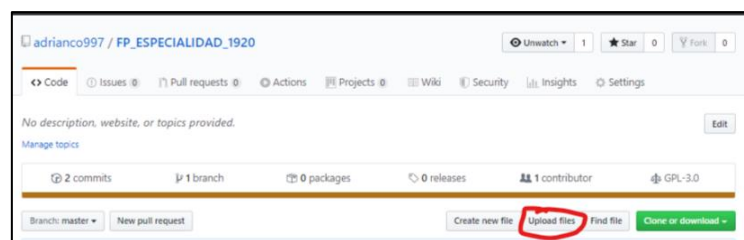
☐ Private: You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README: This will let you immediately clone the repository to your computer.

Add .gitignore: Add a license:

En el repositorio pulsar subir archivos:



Añadir los archivos, un comentario de la subida y guardar:

adrianco997 / FP_ESPECIALIDAD_1920

FP_ESPECIALIDAD_1920 /

Drag files here to add them to your repository
Or choose your files

Commit changes

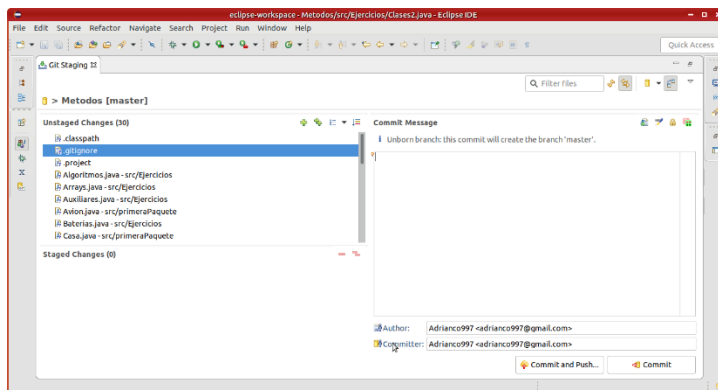
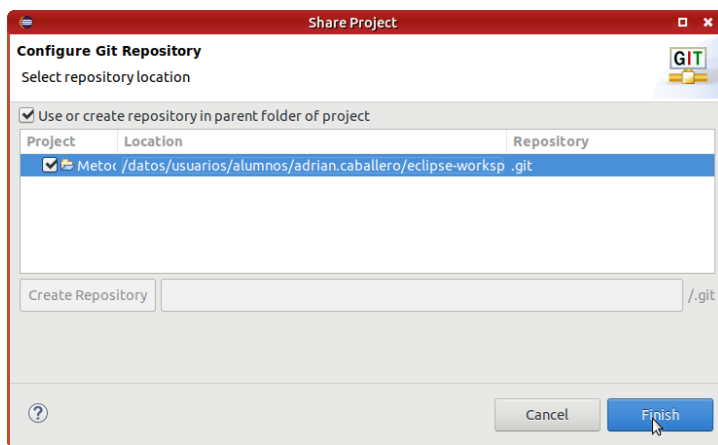
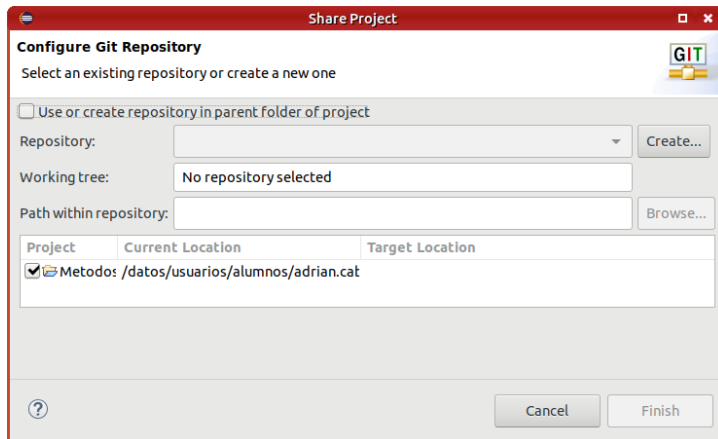
Add files via upload:

Add an optional extended description:

☒ Commit directly to the `main` branch

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Eclipse



GIT Bash

```
$ git config --global user.name "Adrianco997"
$ git config --global user.email "adrianco997@gmail.com"
$ mkdir FP_ESPECIALIDAD_1920
$ cd FP_ESPECIALIDAD_1920
```

1. Crea 3 archivos con cualquier contenido: archivo1.txt, archivo2.txt y archivo3.txt. Estos archivos están contenidos en sus respectivos directorios directorio1, directorio2 y directorio3.

```
$ mkdir directorio1 directorio2 directorio3
$ touch directorio1/archivo1.txt directorio2/archivo2.txt directorio3/archivo3.txt
$ ls -l ./
.:
directorio1/ directorio2/ directorio3/
./directorio1:
archivo1.txt
./directorio2:
archivo2.txt
./directorio3:
archivo3.txt
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    directorio1/
    directorio2/
    directorio3/
nothing added to commit but untracked files present (use "git add" to track)
```

2. Inicializa un repositorio local

```
$ cd ..
$ git init FP_ESPECIALIDAD_1920
$ cd FP_ESPECIALIDAD_1920
```

3. Añade al stage los archivos de los directorios 1 y 2.

```
$ git add directorio1 directorio2
```

4. Comprueba con git status que el stage tiene lo que quieres.

```
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   directorio1/archivo1.txt
    new file:   directorio2/archivo2.txt
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    directorio3/
```

5. Crea un commit con esos dos archivos.

```
$ git commit -m "Añadido texto a 1/1.txt y 2/2.txt"
[master (root-commit) d77bfla] Añadido texto a 1/1.txt y 2/2.txt
 2 files changed, 1 insertion(+)
 create mode 100644 directorio1/archivo1.txt
 create mode 100644 directorio2/archivo2.txt
```

6. Realiza una modificación en el texto del archivo 1/1.txt y otra en el texto del archivo 2/2.txt.

```
$ vim directorio1/archivo1.txt
$ vim directorio2/archivo2.txt
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   directorio1/archivo1.txt
    new file:   directorio2/archivo2.txt
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:   directorio1/archivo1.txt
    modified:   directorio2/archivo2.txt
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    directorio3/
```

7. Añade al stage únicamente las modificaciones del archivo 2/2.txt.

```
$ git add directorio2
```

8. Haz git status e interpreta la información que te da.

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   directorio1/archivo1.txt
```

```
new file:   directorio2/archivo2.txt
```

```
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:   directorio1/archivo1.txt
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
directorio3/
```

9. Crea un commit con las modificaciones que has hecho en el archivo 2/2.txt y con el archivo 3/3.txt

```
$ git commit -m "Añadido texto a 2/2.txt y 3/3.txt"
```

```
On branch master
```

```
Changes not staged for commit:
```

```
modified:   directorio1/archivo1.txt
```

```
Untracked files:
```

```
directorio3/
```

```
no changes added to commit
```

10. Haz git status y comprueba que el stage está vacío y que en el working directory está el cambio del archivo 1/1.txt

```
$ git status
```

```
On branch master
```

```
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:   directorio1/archivo1.txt
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
directorio3/
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

11. Desversiona el archivo 1/1.txt (es decir, dile a Git que no lo versione), pero sin eliminarlo.

```
$ git rm --cached directorio1/archivo1.txt
```

```
rm 'directorio1/archivo1.txt'
```

12. Elimina el archivo 2/2.txt.

```
$ rm directorio2/archivo2.txt
```

13. Haz git status y comprueba que la eliminación de los dos archivos está en el stage.

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git restore --staged <file>..." to unstage)
```

```
deleted:    directorio1/archivo1.txt
```

```
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:    directorio2/archivo2.txt
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
directorio1/
```

```
directorio3/
```

14. Crea un commit que almacene la eliminación de esos dos últimos archivos.

```
$ git commit -m "Eliminados a directorio1/archivo1.txt y directorio2/archivo2.txt"
```

```
[master 00036f6] Eliminados a directorio1/archivo1.txt y directorio2/archivo2.txt
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
delete mode 100644 directorio1/archivo1.txt
```

15. Haz git status y comprueba que todo está como esperas (el archivo 1/1.txt está sin versionar y el stage está sin cambios).

```
$ git status
On branch master
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:    directorio2/archivo2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        directorio1/
        directorio3/

no changes added to commit (use "git add" and/or "git commit -a")
```

16. Configura el repositorio para que ignore la carpeta 1.

```
$ vim ../.git/info/exclude
$ cat ../.git/info/exclude
# git ls-files --others --exclude-from=../.git/info/exclude
# Lines that start with '#' are comments.
# For a project mostly in C, the following would be a good set of
# exclude patterns (uncomment them if you want to use them):
# *.o
# *~
./directorio1/
./directorio1/*
```

17. Haz git status y comprueba la diferencia.

```
$ git status
On branch master
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:    directorio2/archivo2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        directorio1/
        directorio3/

no changes added to commit (use "git add" and/or "git commit -a")
```