

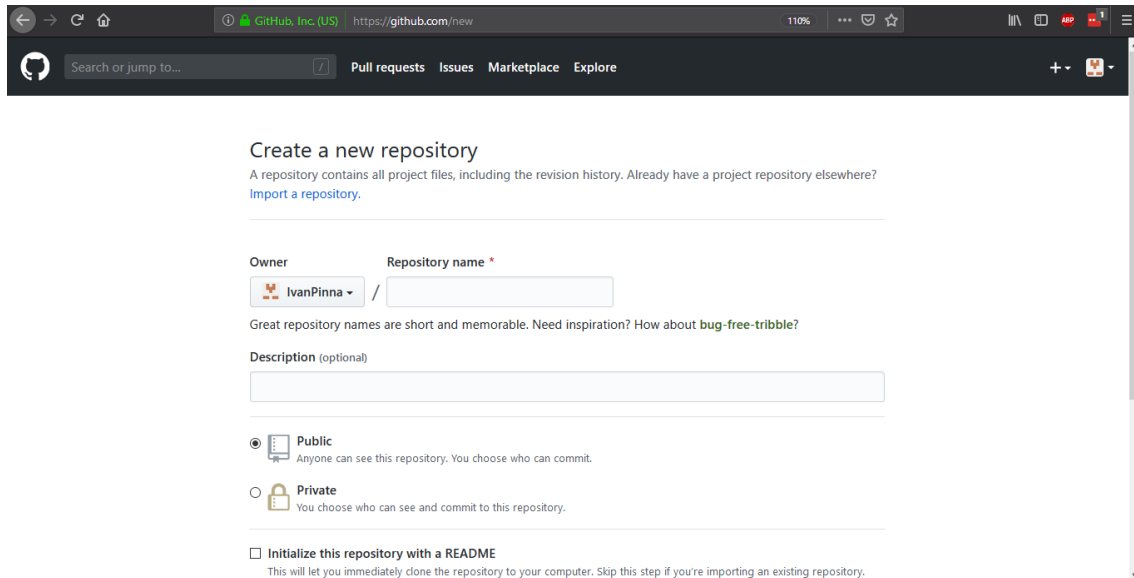
2019

Seminario GitHub

ROCÍO UTRERA RODRIGUEZ
IVÁN PIÑA ARÉVALO
MOISES GUERRERO LÓPEZ

En esta actividad explicaremos como subir un archivo a GitHub así como los comandos necesarios para sincronizar el repositorio, subir archivos y modificarlos.

El encargado de crear el repositorio fue Iván. Iván lo realizó a través de la página web de Github, con la opción “create new repository”



Create a new repository

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

Owner: IvanPinna / Repository name:

Great repository names are short and memorable. Need inspiration? How about [bug-free-tribble](#)?

Description (optional):

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

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

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Después, lo único necesario fue abrir una terminal y teclear

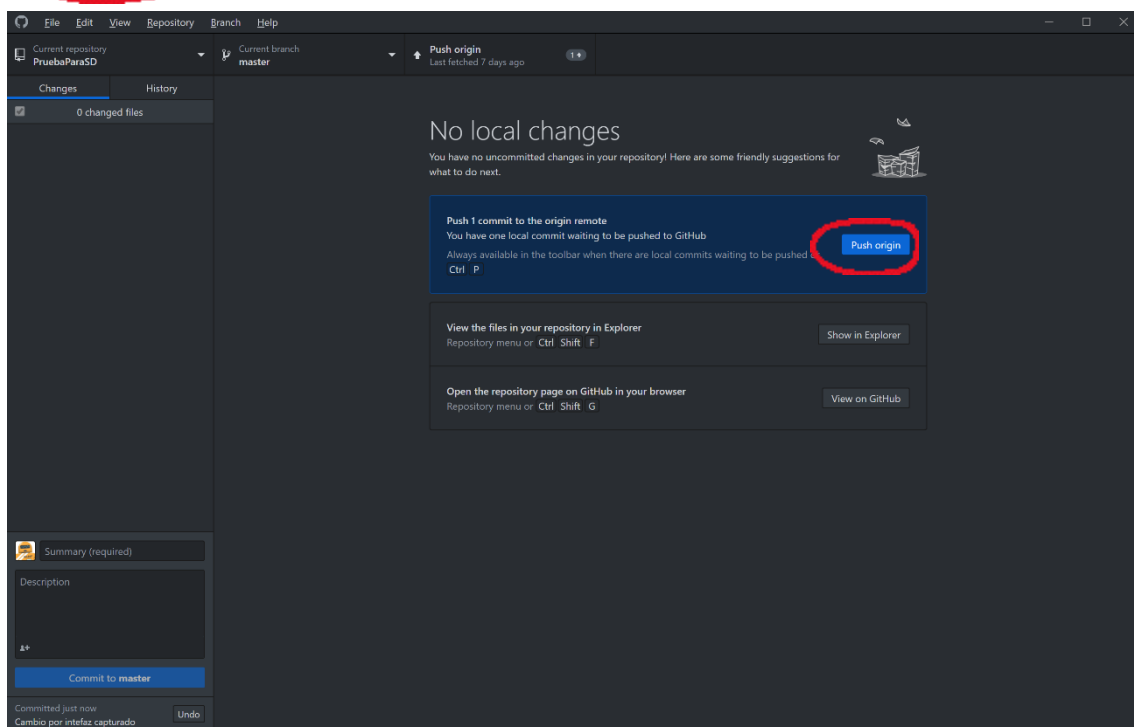
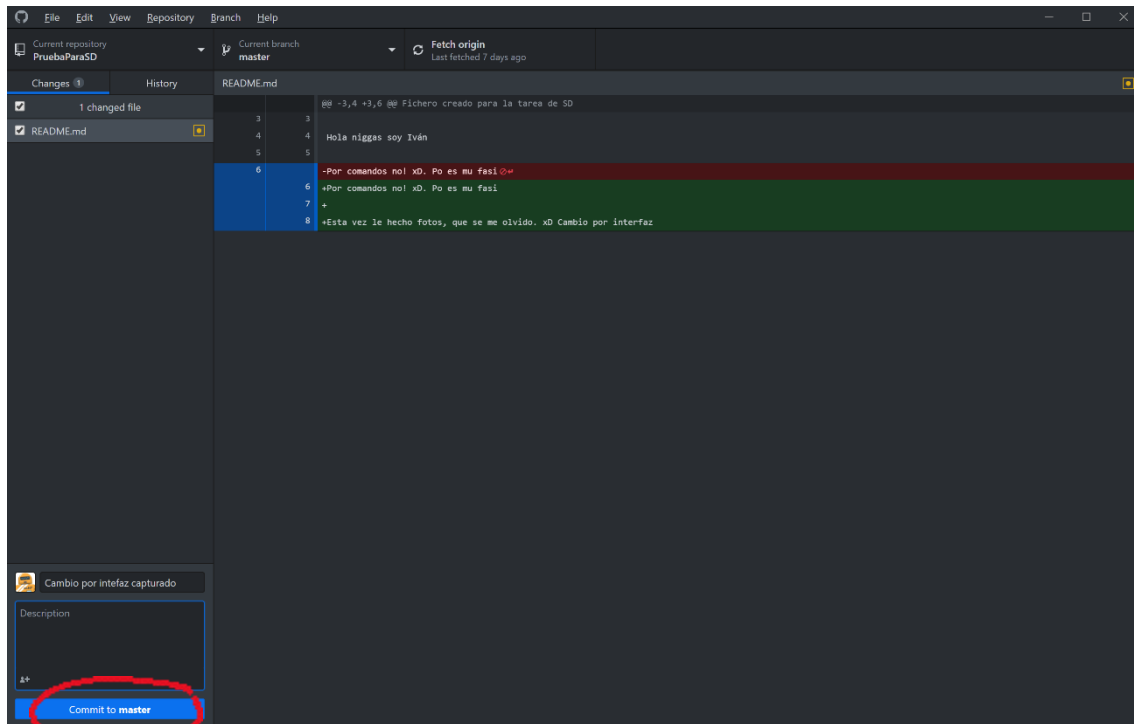
“git clone <https://github.com/IvanPinna/PruebaParaSD>”

De esta manera se replica el repositorio en el ordenador.

Por una parte, Rocío subió archivos con la interfaz gráfica y por comandos. Se explicará primero por la interfaz gráfica.

Una vez instalado la interfaz y creado nuestro repositorio, para añadir algún archivo o modificarlo, solo hay que añadir o modificar en la carpeta local. Luego darle a commit. Y finalmente pulsamos el botón push.

Mostramos las imágenes.



Para la versión de comandos, una vez tenemos el repositorio y los archivos nuevos o modificados, se da el comando `'git status'` para verificarlo (este paso no es obligatorio, pero si recomendado), luego `'git add [nombre de los ficheros]'`, seguido de `'git commit -m`

“Comentario”, y se finaliza con ‘git push’.

```
Windows PowerShell
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git status
On branch master
Your branch is up to date with 'origin/master'.

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

        Commit.png
        Push.png

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git add .\Commit.png
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git add .\Push.png
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   Commit.png
        new file:   Push.png

PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git commit -m "he terminado el curso"
[master a062c5] he terminado el curso
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Commit.png
 create mode 100644 Push.png
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git push
To https://github.com/IvanPina/PruebaParaSD.git
 ! [rejected]        master -> master (fetch first)

hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/IvanPina/PruebaParaSD
   1555f2f..44053dd master -> origin/master
Merge made by the 'recursive' strategy.
 README.md | 4
 1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD> git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 154.16 KiB | 11.51 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/IvanPina/PruebaParaSD.git
   44053dd..a1ffc4a master -> master
PS C:\Users\rutre\OneDrive - alum.uca.es\Universidad\SD 18-19\PruebaParaSD>
```

En esta imagen vemos que al realizar el “push”, nos manda un error. Y es que los compañeros estuvieron realizando cambios en el repositorio, por lo que los archivos locales que tenía no eran los más recientes. Se actualiza el repositorio local con ‘git pull’, y se realiza el “push” para finalizar.

```
Seleccionar C:\Windows\System32\cmd.exe
G:\MI unidad\Junio4*\SD\PruebaParaSD>git rm Borrarr.txt
rm 'Borrarr.txt'

G:\MI unidad\Junio4*\SD\PruebaParaSD>git push
Everything up-to-date

G:\MI unidad\Junio4*\SD\PruebaParaSD>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

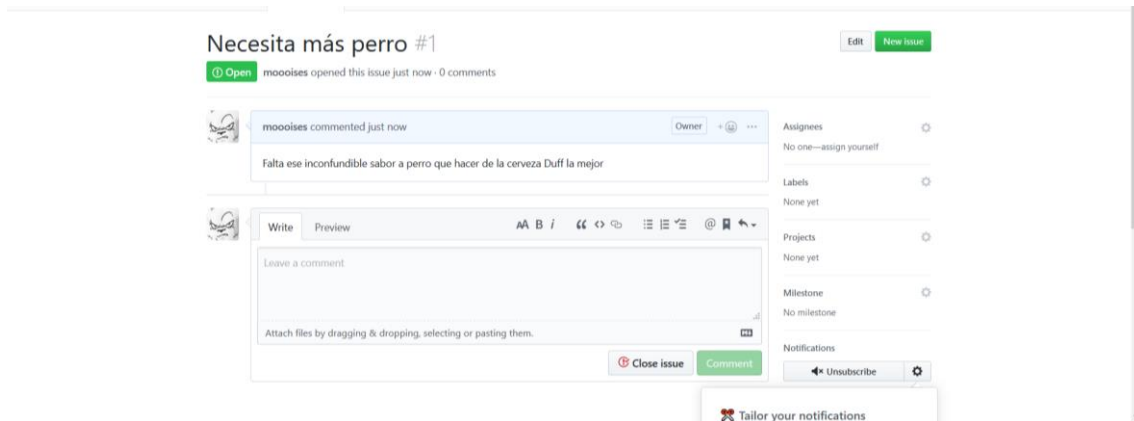
        deleted:   Borrarr.txt

G:\MI unidad\Junio4*\SD\PruebaParaSD>git commit -m "Borrado"
[master a1ffc4a] Borrado
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 Borrarr.txt

G:\MI unidad\Junio4*\SD\PruebaParaSD>git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 213 bytes | 7.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/IvanPina/PruebaParaSD
   b29ed80..a1ffc4a master -> master
G:\MI unidad\Junio4*\SD\PruebaParaSD>
```

Usamos el comando ‘git rm’ para borrar un archivo del repositorio , sin olvidar hacer ‘git commit’ y ‘git push’.

Para tratar las issue, basta con añadirla en el repositorio de GitHub en la pestaña ‘Issue’ y a ‘New Issue’



Finalmente, trataremos el tema de las ramas (en inglés 'branch'). Las ramas nos permiten trabajar paralelamente en un mismo repositorio, sin afectar a los demás usuarios. Tenemos la rama 'master' que es la principal, mas todas las que queramos crear. Para crear una rama se utiliza la orden

`"git branch "nombre_rama" "`

Y para cambiar de una rama a otra

`"git checkout "nombre_rama" "`

```
C:\Windows\System32\cmd.exe

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git checkout ivan
error: pathspec 'ivan' did not match any file(s) known to git

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git branch ivan

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>ddir
"ddir" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 8EFE-EAF1

Directorio de C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD
22/05/2019  12:02    <DIR>          .
22/05/2019  12:02    <DIR>          ..
22/05/2019  11:54             96,266 Commit.png
22/05/2019  12:02             570,083 Memoria GitHub.docx
22/05/2019  12:02             588,798 Memoria GitHub.pdf
22/05/2019  11:54             145,380 Push.png
22/05/2019  11:54              204 README.md
                5 archivos             1,400,731 bytes
                2 dirs  23,605,428,224 bytes libres

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git checkout ivan
Switched to branch 'ivan'

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>
```

Finalmente, una vez hayamos completado el trabajo que realizamos en la rama, podemos fusionarla con la rama master para incorporar en master las mejoras de la rama. Para eso nos situamos en la rama master y ejecutamos la orden:

`"git merge "nombre_rama" "`

Importante, una vez creada la rama es importante notificárselo a github, para eso usamos la orden

“git push --set-upstream origin nombre_rama”

```
C:\Windows\System32\cmd.exe
C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git add README.md

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git commit -m "Modificado readme desde rama ivan"
[ivan c7de916] Modificado readme desde rama ivan
1 file changed, 2 insertions(+)

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git push
fatal: The current branch ivan has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin ivan

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git push --set-upstream origin ivan
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 330 bytes | 165.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'ivan' on GitHub by visiting:
remote:   https://github.com/IvanPinna/PruebaParaSD/pull/new/ivan
remote:
To https://github.com/IvanPinna/PruebaParaSD
 * [new branch]      ivan -> ivan
Branch 'ivan' set up to track remote branch 'ivan' from 'origin'.

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>git merge ivan
Updating df174ab..c7de916
Fast-forward
 README.md | 2 ++
1 file changed, 2 insertions(+)

C:\Users\Ivan\Documents\Repositorios GitHub\PruebaParaSD>
```