

Actividad Evaluable

Módulo	Servicios de red	
Nombre y Apellidos:	Daniel Santiso Candel	

FECHA DE ENTREGA: --/--/21

Objetivos

El objetivo de esta actividad individual es la de repasar, asentar y adquirir un mayor conocimiento de lo impartido en clase.

El archivo debe nombrarse como:
nombre_apellido1_apellido2.PDF

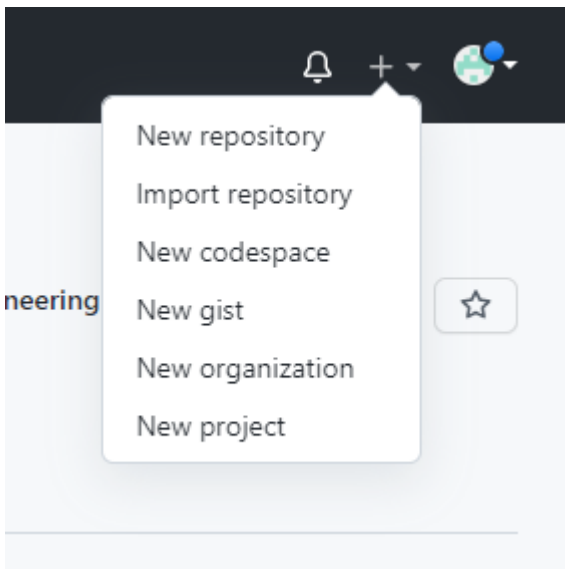
Ej: Belen_Vargas_Sabater.pdf

Cuestiones

EJERCICIO 1: **Práctica:** Realizar una guía práctica a modo de tutorial, sobre el uso de GitHub, algunos de los elementos que debe cubrir son:

- Creación de repositorio y conexión con repositorio personal de gitHub mediante terminal con ejemplos
- Push Y pulls a los repositorios desde el terminal con ejemplos.
- Cada miembro del equipo debe realizar cambios en el repositorio propiedad de uno de los miembros, documenta el proceso.

Para realizar esta práctica, será necesario tener Git instalado y una cuenta de GitHub, para ello tendremos que ir a la página oficial de

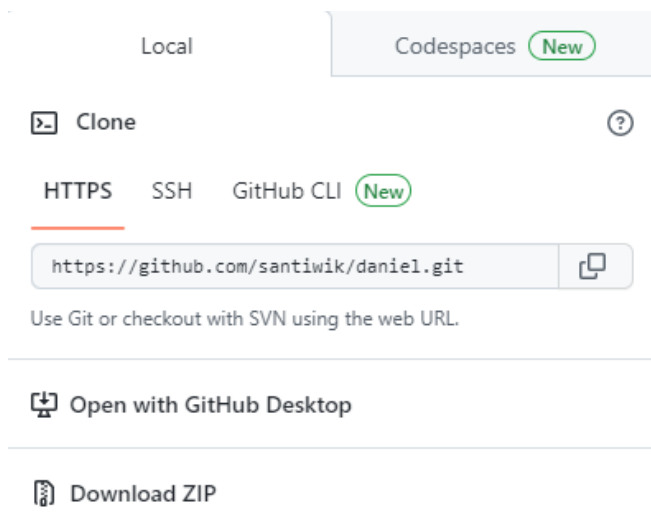


GitHub y crearla, una vez creada, en cualquier página, encontraremos en la esquina superior derecha un + en donde podremos crear varias cosas, sin embargo, nos interesa crear un repositorio.

Para crearlo simplemente tenemos que poner un nombre para el repositorio, y en caso de querer ponerlo privado deberemos seleccionar esto en la página. Yo le pondré el nombre de daniel, es recomendable añadirle un README.txt para que el repositorio ya

tenga archivos.

Para vincular este repositorio con tu repositorio Git personal en tu ordenador, tendremos que copiar el link de tu repositorio.



Y posteriormente tendremos que ir a nuestro ordenador con el Git personal, y poner el comando en modo super usuario: `git remote add [nombre_repositorio] [link_repositorio]`, poniendo el nombre al repositorio nos ahorramos tener que poner siempre el link entero en los comandos que quieras utilizar

```
root@daniel:/home/daniel# git remote add daniel https://github.com/santiwik/daniel.git_
```

Para comprobar que se ha vinculado correctamente el repositorio, podremos el comando: `git remote -v`

```
root@daniel:/home/daniel# git remote -v
daniel https://github.com/santiwik/hello-world2 (fetch)
daniel https://github.com/santiwik/hello-world2 (push)
root@daniel:/home/daniel#
```

Para enviar las branches a GitHub, es necesario hacer un push, para ello tendremos que hacer una token antes para generar la contraseña que nos pedirá al hacer el push, para crear el token tendremos que ir a los ajustes de la cuenta e ir a <>developer settings, posteriormente iremos a las personal Access tokens y crearemos una clásica y nos saldrá este menú en donde tendremos que seleccionar los repositorios y podremos configurar el tiempo en el que se expirará el token.

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

hello world

What's this token for?

Expiration *

30 days


The token will expire on Fri, Feb 17 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

- | | |
|---|--------------------------------------|
| <input checked="" type="checkbox"/> repo | Full control of private repositories |
| <input checked="" type="checkbox"/> repo:status | Access commit status |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status |
| <input checked="" type="checkbox"/> public_repo | Access public repositories |
| <input checked="" type="checkbox"/> repo:invite | Access repository invitations |
| <input checked="" type="checkbox"/> security_events | Read and write security events |

Al crearla, nos saldrá un código el cual copiaremos, este código tendremos que guardarlo.

✓ ghp_8P1HmNu4fhY0iGuSmO63AKYomQFhye14zfwk 

Delete

Ahora iremos a nuestro ordenador y pondremos el comando: `git push --all [nombre_repo]` enviando todas nuestras ramas incluida la master a github.













```
root@daniel:/home/daniel/Escritorio# git push --all daniel
```

Sin embargo, nos pedirá el nombre del usuario y su contraseña, en su contraseña pondremos el código del token.

La contraseña será invisible por ello hay que tener mucho cuidado al ponerla. Con esto ya tendremos todo el trabajo vinculado en GitHub.

```
Username for 'https://github.com': daniel
Password for 'https://daniel@github.com':
```

```
Enumerando objetos: 12, listo.
Contando objetos: 100% (12/12), listo.
Comprimiendo objetos: 100% (4/4), listo.
Escribiendo objetos: 100% (12/12), 884 bytes | 884.00 KiB/s, listo.
Total 12 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/santiwik/daniel.git
* [new branch]      hola -> hola
* [new branch]      master -> master
```

Your branches			
hola	Updated 42 minutes ago by santiwik	1 4	 New pull request  
master	Updated 48 minutes ago by santiwik	1 2	 New pull request  
Active branches			
hola	Updated 42 minutes ago by santiwik	1 4	 New pull request  
master	Updated 48 minutes ago by santiwik	1 2	 New pull request  

Un pull es traer el trabajo de una Branch a tu ordenador a través del comando pull, esto consiste en el siguiente comando: `git pull [nom_rep] [nom_branch]`

```
root@daniel:/home/daniel/Escritorio# git pull daniel main
Desde https://github.com/santiwik/daniel
* branch      main      -> FETCH_HEAD
```

Así tendría ya lo que había en la Branch main de GitHub.

Para hacer el tercer ejercicio creare un nuevo repositorio (repositorio2) para reubicarme. Como solo soy uno solo haré yo los cambios en el repositorio.

Una vez hecho git init en una carpeta creada, hice el primer commit para tener una pequeña copia de seguridad sin nada. Al hacer esto cree un txt llamado prueba, lo edite con el comando nano poniendo hola mundo, y vuelvo a hacer un commit para guardar el contenido no sin antes haberlo añadido a git con el comando git add. Vinculare el repositorio remotamente como se ha explicado anteriormente.

```
root@holasoydani:/home/daniel/hola# git remote add repositorio2 https://github.com/santiwik/repositorio2.git
```

Después de haberlo vinculado haré un push para subir el segundo commit.

```
root@holasoydani:/home/daniel/hola# git push --all repositorio2
Username for 'https://github.com': santiwik
Password for 'https://santiwik@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 222 bytes | 222.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/santiwik/repositorio2.git
 * [new branch]      master -> master
```

Posteriormente haré una Branch llamada segunda_branch y modificare el txt poniendo adiós mundo y crearé otro (prueba2.txt) poniendo “hola mundo otra vez”. Para unir el trabajo de la branch principal con tu trabajo es necesario hacer un pull, mergear la segunda branch en la branch principal y hacer un push de nuevo en donde se nos subirá la branch principal actualizada y la segunda branch en caso de querer volver a trabajar con ella que como mergeamos tienen el mismo contenido, modificando el contenido del primer commit en la branch principal.

```

root@holasoydani:/home/daniel/hola# git branch
* master
  segunda_branch
root@holasoydani:/home/daniel/hola# ls
prueba.txt
root@holasoydani:/home/daniel/hola# git pull repositorio2 master
From https://github.com/santiwik/repositorio2
 * branch                master      -> FETCH_HEAD
Already up to date.
root@holasoydani:/home/daniel/hola# git merge segunda_branch
Updating 93bada1..7d08175
Fast-forward
 prueba.txt | 2 +-
 prueba2.txt | 2 ++
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 prueba2.txt
root@holasoydani:/home/daniel/hola# ls
prueba2.txt prueba.txt

```

adios
mundo

hola mundo
otra vez

Estas imágenes ya son una prueba de que hemos hecho bien el trabajo ahora solo hace falta

hacer el push para enviar los datos actualizados.

```

root@holasoydani:/home/daniel/hola# git push --all repositorio2
Username for 'https://github.com': santiwik
Password for 'https://santiwik@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 315 bytes | 315.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/santiwik/repositorio2.git
 93bada1..7d08175 master -> master
 * [new branch]      segunda_branch -> segunda_branch

```

Vamos a GitHub para ver si se ha subido correctamente.

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with 'master' selected, '2 branches', and '0 tags'. Buttons for 'Go to file', 'Add file', and 'Code' are visible. A warning box states 'Your master branch isn't protected' with a 'Protect this branch' button. Below this, a commit by 'santiwik' is shown with the message 'primer commit del branch'. The commit details table lists two files: 'prueba.txt' and 'prueba2.txt', both from the 'primer commit del branch' and committed '13 minutes ago'.

File	Commit Message	Time
prueba.txt	primer commit del branch	13 minutes ago
prueba2.txt	primer commit del branch	13 minutes ago