

# TRABAJO PRACTICO ESPECIAL

## GIT-GITHUB

TECNOLOGIA DE LA INFORMACION EN LAS ORGANIZACIONES

TUDAI - FACULTAD DE CIENCIAS EXACTAS - UNICEN

### Integrantes:

- Asi, Milagros
- González Toraf, Abril
- Moller, Cristian
- Silva, Florencia
- Silva, Micaela

## INDICE

UNA PEQUEÑA INTRODUCCION .....	1
¿Qué es GitHub? .....	1
¿Para qué sirve? .....	1
¿Qué herramientas proporciona? .....	1
¿Qué uso le daremos?.....	1
CREACION DEL REPOSITORIO .....	2
COMANDOS: GIT STATUS, GIT ADD, GIT COMMIT, GIT PUSH .....	3
COMANDO: GIT LOG .....	4
COMANDOS: GIT BRANCH, GIT CHECKOUT .....	5
ISSUES Y MILESTONES .....	6
MERGE PULL REQUEST .....	7
CONCLUSION.....	8
Link repositorio GitHub .....	8

# UNA PEQUEÑA INTRODUCCION

## ¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

## ¿Para qué sirve?

GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

## ¿Qué herramientas proporciona?

Una wiki para el mantenimiento de las distintas versiones de las páginas.

Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.

Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.

Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

## ¿Qué uso le daremos?

Crearemos una cuenta gratuita y comenzaremos a crear repositorios de código, para lograr que el grupo trabaje colaborativamente. Intentando fortalecer nuestros vínculos y comunicación.

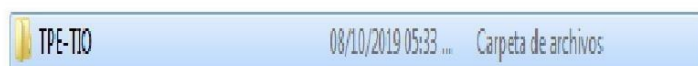
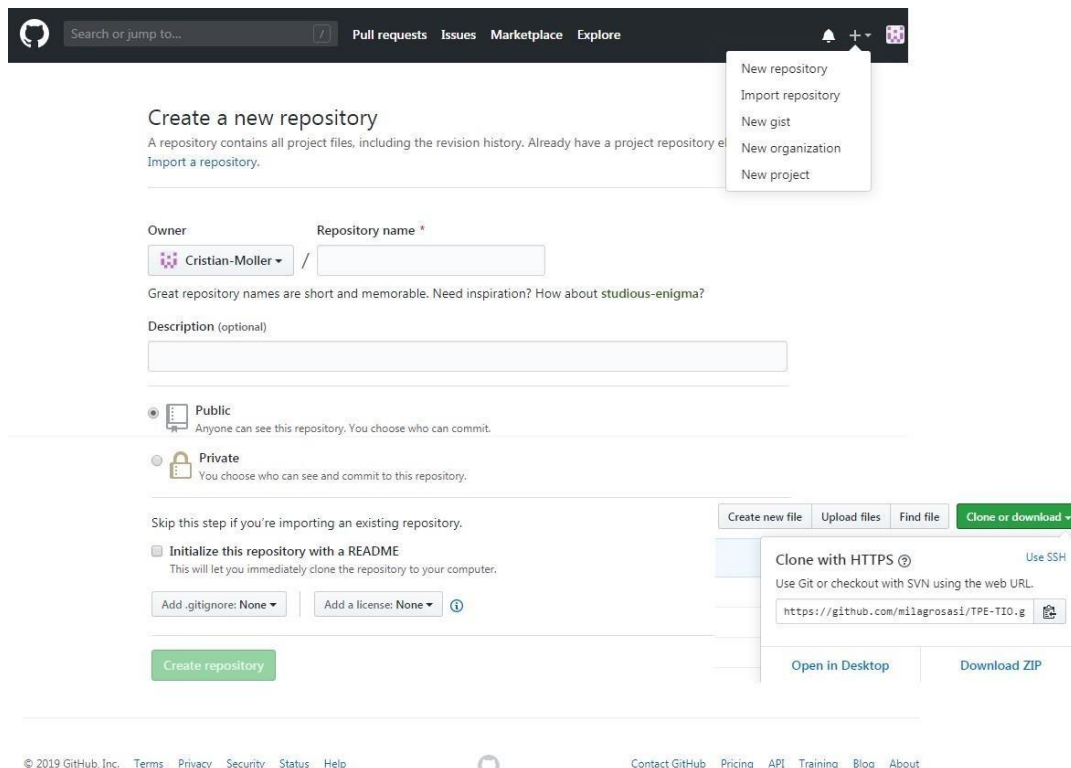
# CREACION DEL REPOSITORIO

1º Iniciamos el TP con la creación del Repositorio Nuevo en la plataforma GITHUB (una vez que teníamos creados nuestros respectivos usuarios).

Le introducimos nombre al repositorio, la creación fue pública, sin extensión y sin licencia.

2º En nuestro escritorio o donde quisiéramos clonar el repositorio, creamos la Carpeta GIT, abrimos la consola de GIT, puede ser por ejemplo Git Bash o powershell en Windows 10.

3º Luego copiamos el enlace para Clonar dicho repositorio. Con el comando **git clone \*URL del repositorio\***, en la consola, clonamos el repositorio dentro de dicha carpeta, para poder comenzar a trabajar en la misma. Este comando además de cumplir la función de “copiarnos” el repositorio en nuestra pc también copia las ramas que estén creadas para poder trabajar en ellas en caso de que sea necesario.



## COMANDOS:

### GIT STATUS, GIT ADD, GIT COMMIT, GIT PUSH

Al iniciar la documentación de los procedimientos, se creó la carpeta documentación, por lo que se debió hacer en **GIT BASH**:

1º Introducir el comando **git status**: el cual nos indica en rojo las carpetas modificadas.

2º Con **git add .** : agregamos los archivos.

3º Con **git commit -m "mensaje"** confirmamos los cambios con el detalle entre comillas.

4º Por último con **git push origin \*nombre\_rama\*** (en este caso es master): subimos los cambios realizados a nuestro repositorio remoto.

```
Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   documentacion/TPE-TIO.docx

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

        documentacion/~WRL0003.tmp

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

Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (master)
$ git add .

Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (master)
$ git commit -m "creacion de repositorio + clone"
[master 4cc42d9] creacion de repositorio + clone
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 documentacion/~WRL0003.tmp

Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (master)
$ git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 4.91 KiB | 93.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/milagrosasi/TPE-TIO.git
 07c152b..4cc42d9 master -> master
```

## COMANDO: GIT LOG

El mismo nos lista los diferentes commits realizados sobre el repositorio creado.

Cada commit realizado tendrá un id, un Autor y una fecha de modificación.

Estos datos identifican el commit y nos permitirán volver un cambio en el caso de ser necesario.

```
Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (master)
$ git log
commit e6aedc52a0b5dbae0727ccc2a93443b7e0a4c9c5 (HEAD -> master, origin/master)
Author: CristianMoller <moller cristian@gmail.com>
Date: Thu Oct 10 23:44:57 2019 -0300

    como commiteamos en repositorio

commit 4cc42d9fa7996542f1227ff74f2ee75ba2be91aa
Author: CristianMoller <moller cristian@gmail.com>
Date: Thu Oct 10 23:21:21 2019 -0300

    creacion de repositorio + clone

commit 07c152bbb211b561b2eedf0ab596f550659efbb1
Author: CristianMoller <moller cristian@gmail.com>
Date: Thu Oct 10 23:15:19 2019 -0300

    creacion de repositorio + clone

commit 1716dd554bb3c3035294afc55c2246b5d732e3d9
Author: CristianMoller <moller cristian@gmail.com>
Date: Tue Oct 8 17:46:29 2019 -0300

    carpeta para documentar

commit 5190c6e4830cb732dca21f0aaf4d1e032a39b48f
Author: CristianMoller <moller cristian@gmail.com>
Date: Tue Oct 8 17:24:15 2019 -0300

    archivos para practicar comandos git
```

## COMANDOS:

### GIT BRANCH, GIT CHECKOUT

El comando `git checkout -b "nombre_rama"` creará una nueva rama y nos posicionará directamente en ella para poder trabajar en paralelo a la rama **master**, para dejar esta última como versión estable en el proyecto.

Para poder cambiar de rama utilizaremos el comando `git checkout "nombre_rama"`.

Luego se puede utilizar el comando `git branch` para verificar el correcto cambio de rama, este comando listará las ramas del proyecto y mostrará en verde y con un asterisco en la rama que estamos ubicados.

Una vez realizados los cambios guardaremos y confirmaremos los cambios realizados, luego los subiremos al proyecto desde esta nueva rama.

Las ramas son muy útiles para trabajo en equipo y simultaneo, donde cada usuario tendrá su propia rama y podrá trabajar sin riesgo de entorpecer el trabajo de sus compañeros.

```
Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (master)
$ git branch
  cristian
* master

Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (master)
$ git checkout cristian
Switched to branch 'cristian'

Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (cristian)
$ git branch
* cristian
  master

Administrador@WIN-EU1IGSRFRDN MINGW32 ~/Desktop/git/TPE-TIO (cristian)
$
```

## ISSUES Y MILESTONES

☐ 1 Open ✓ 5 Closed


Author ▾ Labels ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾


☐  Actualizar informe  
#6 opened 30 minutes ago by abrilgonzaleztoraf

Las issues son “publicaciones” que se hacen dentro del repositorio para mostrar problemas o solicitar que los miembros del equipo ayuden con a resolverlas. Estas se crean dentro de la web y quedan abiertas hasta que se encuentra una solución, luego la issue se puede cerrar.


Las Milestone son “cajas” contenedoras de issues donde se pueden poner las issue que estén relacionadas al mismo tema.

### Actualizar informe #6



 Closed abrilgonzaleztoraf opened this issue 33 minutes ago · 1 comment



abrilgonzaleztoraf commented 33 minutes ago Collaborator + 😊 ...  
El informe debe ser actualizado en la brevedad



abrilgonzaleztoraf commented now Author Collaborator + 😊 ...  
Done


  abrilgonzaleztoraf closed this now




## MERGE PULL REQUEST

Para hacer el Merge fuimos a la página con la rama ya creada y commiteada y desde la misma hicimos el pull request. Una vez que git verifica que el código a mergear no genera conflictos con el código que hay en master, te deja hacer el merge y el código queda todo unido en master.


### Solucion issue #6 informe actualizado #7

 milagrosasi wants to merge 1 commit into `master` from `abril`


Conversation 0 Commits 1 Checks 0 Files changed 3

 milagrosasi commented now Owner + 😊 ...

No description provided.


 Solucion issue #6 informe actualizado 48407d5

Add more commits by pushing to the `abril` branch on milagrosasi/TPE-TIO.


 **This branch has no conflicts with the base branch**  
Merging can be performed automatically.

[Merge pull request](#) You can also open this in GitHub Desktop or view command line instructions.


### Solucion issue #6 informe actualizado #7


 milagrosasi merged 1 commit into `master` from `abril` now


Conversation 0 Commits 1 Checks 0 Files changed 3

 milagrosasi commented 1 minute ago Owner + 😊 ...

No description provided.

 Solucion issue #6 informe actualizado 48407d5

 milagrosasi merged commit `f0cda0d` into `master` now Revert

 **Pull request successfully merged and closed**  
You're all set—the `abril` branch can be safely deleted. Delete branch

## CONCLUSION

Este trabajo nos sirvió para poder conocer la herramienta Git - GitHub y aprender a manejarla, crear repositorios o trabajar con repositorios ya existentes. La herramienta es muy útil a la hora de continuar con el proyecto, ya que por cuestiones de tiempo o espacio no siempre es posible estar todos reunidos en un mismo lugar físico; y uno puede seguir trabajando en forma colaborativa, simplemente bajando las actualizaciones, de los avances ya realizados por nuestros compañeros.

Link repositorio GitHub

<https://github.com/milagrosasi/TPE-TIO/>