

Trabajo Practico 2 – GIT



Alumno: Patricio Sussini Guanziroli

Materia: Programación 1

Profesor: Bruselario, Sebastián

Tutora: Gubiotti, Florencia

Índice de Actividades:

1. Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :
 - a. ¿Qué es GitHub?
 - b. ¿Cómo crear un repositorio en GitHub?
 - c. ¿Cómo crear una rama en Git?
 - d. ¿Cómo cambiar a una rama en Git?
 - e. ¿Cómo fusionar ramas en Git?
 - f. ¿Cómo crear un commit en Git?
 - g. ¿Cómo enviar un commit a GitHub?
 - h. ¿Qué es un repositorio remoto?
 - i. ¿Cómo agregar un repositorio remoto a Git?
 - j. ¿Cómo empujar cambios a un repositorio remoto?
 - k. ¿Cómo tirar de cambios de un repositorio remoto?
 - l. ¿Qué es un fork de repositorio?
 - m. ¿Cómo crear un fork de un repositorio?
 - n. ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
 - o. ¿Cómo aceptar una solicitud de extracción?
 - p. ¿Qué es un etiqueta en Git?
 - q. ¿Cómo crear una etiqueta en Git?
 - r. ¿Cómo enviar una etiqueta a GitHub?
 - s. ¿Qué es un historial de Git?
 - t. ¿Cómo ver el historial de Git?
 - u. ¿Cómo buscar en el historial de Git?
 - v. ¿Cómo borrar el historial de Git?
 - w. ¿Qué es un repositorio privado en GitHub?
 - x. ¿Cómo crear un repositorio privado en GitHub?
 - y. ¿Cómo invitar a alguien a un repositorio privado en GitHub?
 - z. ¿Qué es un repositorio público en GitHub?
 - aa. ¿Cómo crear un repositorio público en GitHub?
 - bb. ¿Cómo compartir un repositorio público en GitHub?
2. Realizar actividad en GitHub.
3. Realizar otra actividad en GitHub.

a. ¿Qué es GitHub?

Es una plataforma de alojamiento de código que permite control de versiones y colaboración con Git.

b. ¿Cómo crear un repositorio en GitHub?

1. Iniciar sesión en GitHub.
2. Ir a "Repositories" y hacer clic en "New".
3. Completar nombre, visibilidad y hacer clic en "Create repository".

c. ¿Cómo crear una rama en Git?

`git branch nombre_rama`

d. ¿Cómo cambiar a una rama en Git?

`git checkout nombre_rama` o `git switch nombre_rama`

e. ¿Cómo fusionar ramas en Git?

1. Cambiar a la rama principal: `git checkout main`
2. Ejecutar: `git merge nombre_rama`

f. ¿Cómo crear un commit en Git?

1. Agregar cambios: `git add .`
2. Confirmar cambios: `git commit -m "Mensaje"`

g. ¿Cómo enviar un commit a GitHub?

`git push origin + nombre de la rama`

h. ¿Qué es un repositorio remoto?

Es una versión alojada en la nube de un repositorio local.

i. ¿Cómo agregar un repositorio remoto a Git?

`git remote add origin URL_del_repositorio`

j. ¿Cómo empujar cambios a un repositorio remoto?

`git push origin nombre_rama`

k. ¿Cómo tirar de cambios de un repositorio remoto?

`git pull origin nombre_rama`

l. ¿Qué es un fork de repositorio?

Es una copia de un repositorio en una cuenta propia para hacer cambios sin afectar el original.

m. ¿Cómo crear un fork de un repositorio?

1. Ir al repositorio en GitHub.
2. Hacer clic en "Fork".

n. ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

1. Subir cambios a la rama en el fork.
2. Ir a "Pull Requests" y crear una nueva.

o. ¿Cómo aceptar una solicitud de extracción?

1. Revisar el PR en GitHub.
2. Hacer clic en "Merge Pull Request".

p. ¿Qué es una etiqueta en Git?

Es un marcador en el historial de Git usado para versiones.

q. ¿Cómo crear una etiqueta en Git?

`git tag -a v1.0 -m "Versión 1.0"`

r. ¿Cómo enviar una etiqueta a GitHub?

`git push origin v1.0`

s. ¿Qué es un historial de Git?

Es el registro de cambios realizados en un repositorio.

t. ¿Cómo ver el historial de Git?

`git log`

u. ¿Cómo buscar en el historial de Git?

`git log --grep="palabra_clave"`

v. ¿Cómo borrar el historial de Git?

No se puede borrar, pero se puede resetear con `git rebase` o `git reset`.

w. ¿Qué es un repositorio privado en GitHub?

Un repositorio accesible solo para usuarios autorizados.

x. ¿Cómo crear un repositorio privado en GitHub?

1. Crear un nuevo repositorio.
2. Seleccionar "Private" antes de crearlo.

y. ¿Cómo invitar a alguien a un repositorio privado en GitHub?

1. Ir a "Settings" luego "Manage access".
2. Hacer clic en "Invite a collaborator".

z. ¿Qué es un repositorio público en GitHub?

Un repositorio accesible para cualquier usuario.

aa. ¿Cómo crear un repositorio público en GitHub?

1. Crear un nuevo repositorio.
2. Seleccionar "Public" antes de crearlo.

bb. ¿Cómo compartir un repositorio público en GitHub?
Compartiendo la URL del repositorio.

Actividad 2 con GitHub:

- Creo un repo, le doy nombre, elijo que sea público y lo inicializo con un archivo.

The image displays two screenshots from the GitHub web interface. The top screenshot shows the 'Create a new repository' form. The 'Repository name' field is filled with 'Repo_Actividad_2', and the 'Owner' is 'sussiniguaziroli'. The 'Public' radio button is selected. The 'Add a README file' checkbox is checked. The 'Create repository' button is visible at the bottom right. The bottom screenshot shows the newly created repository page for 'Repo_Actividad_2'. It displays the repository name, owner, and a commit message: 'Agrego Archivo .txt para inicializar el repo'. A file named 'Inicializacion_Repo.txt' is listed. The 'Add a README' button is prominent in the center. The right sidebar shows repository statistics: 0 stars, 1 watching, and 0 forks.

```

MINGW64/c/Users/paddy/Desktop/carpeta chitera/Repo_Actividad_2
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git add .

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git commit -m "Agregando mi-archivo.txt"
[main a308bdd] Agregando mi-archivo.txt
1 file changed, 1 insertion(+)
create mode 100644 mi-archivo.txt

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/sussiniguanziroli/Repo_Actividad_2
beaa331..a308bdd main -> main

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ |

```

```

MINGW64/c/Users/paddy/Desktop/carpeta chitera/Repo_Actividad_2
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git branch nueva_rama

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git add .

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git commit -m "Agrego cambios a la nueva rama"
[main bebd5f9] Agrego cambios a la nueva rama
1 file changed, 1 insertion(+)
create mode 100644 nueva_rama_archivo_modificacion.txt

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 358 bytes | 358.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sussiniguanziroli/Repo_Actividad_2
a308bdd..bebd5f9 main -> main

```

```

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (main)
$ git checkout nueva_rama
Switched to branch 'nueva_rama'

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (nueva_rama)
$ git add .

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (nueva_rama)
$ git commit -m "agrego archivo a la nueva rama"
[nueva_rama a49080e] agrego archivo a la nueva rama
1 file changed, 1 insertion(+)
create mode 100644 nueva_rama_test.txt

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (nueva_rama)
$ git push
fatal: The current branch nueva_rama has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin nueva_rama

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/carpeta chitera/Repo_Actividad_2 (nueva_rama)
$ git push origin nueva_rama
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 350 bytes | 350.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'nueva_rama' on GitHub by visiting:
remote:   https://github.com/sussiniguanziroli/Repo_Actividad_2/pull/new/nueva_rama
remote:
To https://github.com/sussiniguanziroli/Repo_Actividad_2
* [new branch] nueva_rama -> nueva_rama

```

Actividad 3 con GitHub:

Creo el repo y le asigno ese nombre.

Create a new repository

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

Required fields are marked with an asterisk ().*

Owner *

sussiniguanziroli

Repository name *

ejercicio_conflicto

ejercicio_conflicto is available.

Great repository names are short and memorable. Need inspiration? How about [sturdy-guide](#) ?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

① You are creating a public repository in your personal account.

Create repository

Clono dicho repo en mi escritorio:

```
MINGW64:/c/Users/paddy/Desktop
paddy@DESKTOP-SETUP MINGW64 ~/Desktop
$ git clone https://github.com/sussiniguanziroli/ejercicio_conflicto
Cloning into 'ejercicio_conflicto'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

paddy@DESKTOP-SETUP MINGW64 ~/Desktop
$
```

Agrego la rama feature_branch y creo un conflicto a propósito.

```
MINGW64: c:/Users/paddy/Desktop/ejercicio_conflicto
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main)
$ git branch feature_branch

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main)
$ git checkout feature_branch
Switched to branch 'feature_branch'

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (feature_branch)
$ git add README.md

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (feature_branch)
$ git commit -m "Agregue una linea en feature_branch"
[feature_branch 3e6705b] Agregue una linea en feature_branch
1 file changed, 3 insertions(+), 1 deletion(-)

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (feature_branch)
$
```

Ahora vuelvo a main y creo más conflicto a propósito en esa rama.

```
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (feature_branch)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

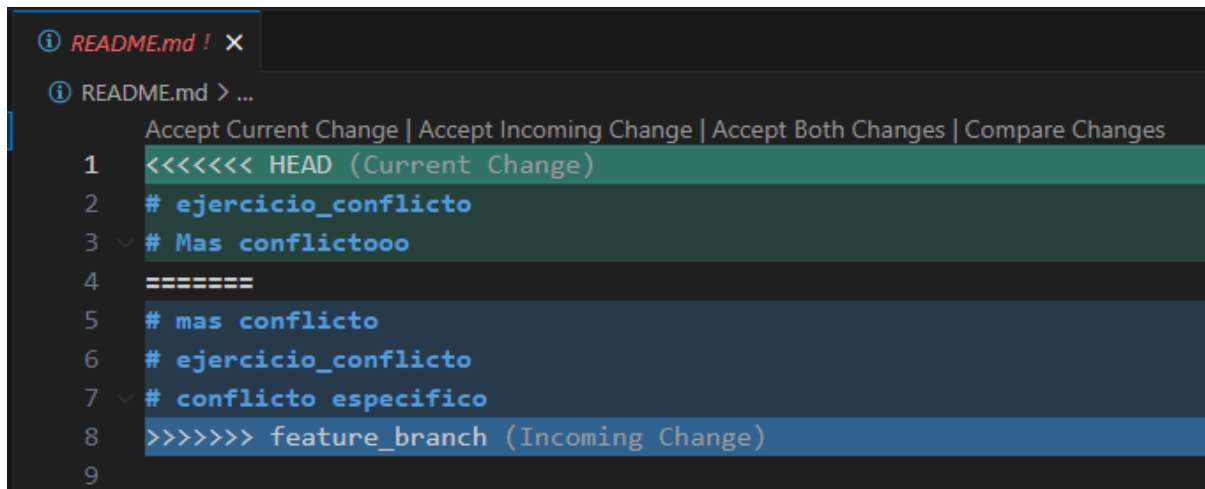
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main)
$ git add README.md

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main)
$ git commit -m "Agregue un conflicto a proposito en la rama main"
[main d2b7c57] Agregue un conflicto a proposito en la rama main
1 file changed, 2 insertions(+), 1 deletion(-)
```

No me deja hacer merge de main con f._branch por el conflicto generado a propósito.

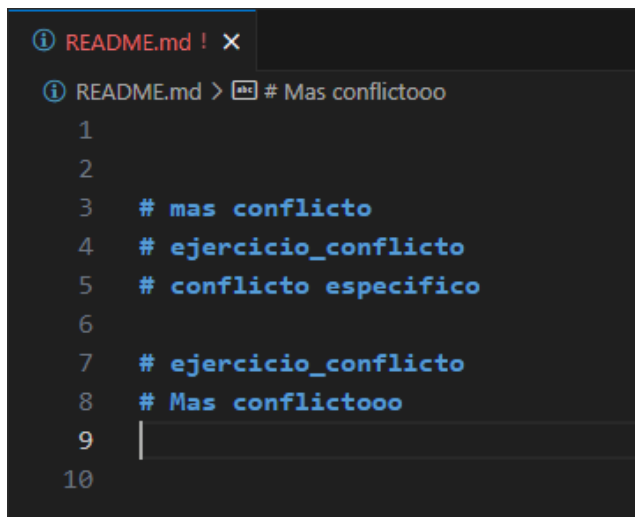
```
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main)
$ git merge feature_branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```


Nueva vista el archivo README.md



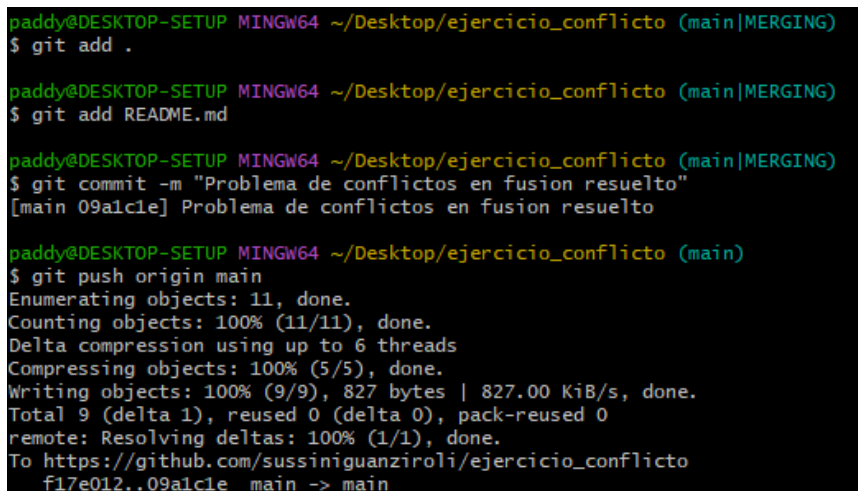
```
1 <<<<<< HEAD (Current Change)
2 # ejercicio_conflicto
3 # Mas conflictooo
4 =====
5 # mas conflicto
6 # ejercicio_conflicto
7 # conflicto especifico
8 >>>>>> feature_branch (Incoming Change)
9
```

Soluciono los conflictos moviendo algún markdown mas abajo usando otras líneas.



```
1
2
3 # mas conflicto
4 # ejercicio_conflicto
5 # conflicto especifico
6
7 # ejercicio_conflicto
8 # Mas conflictooo
9
10
```

Termino con el commit y fusiono las ramas, concluyendo el ejercicio.



```
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main|MERGING)
$ git add .

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main|MERGING)
$ git add README.md

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main|MERGING)
$ git commit -m "Problema de conflictos en fusion resuelto"
[main 09a1c1e] Problema de conflictos en fusion resuelto

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 827 bytes | 827.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/sussiniguanzirol/ejercicio_conflicto
f17e012..09a1c1e main -> main
```

Subo también la rama de características.

```
paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (main)
$ git checkout feature_branch
Switched to branch 'feature_branch'

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (feature_branch)
$ git add .

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (feature_branch)
$ git commit -m "Subo tambien la rama de caracteristicas"
On branch feature_branch
nothing to commit, working tree clean

paddy@DESKTOP-SETUP MINGW64 ~/Desktop/ejercicio_conflicto (feature_branch)
$ git push origin feature_branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature_branch' on GitHub by visiting:
remote:   https://github.com/sussiniganziroli/ejercicio_conflicto/pull/new/feature_branch
remote:
To https://github.com/sussiniganziroli/ejercicio_conflicto
 * [new branch]      feature_branch -> feature_branch
```

Queda registrado en GitHub el conflicto resuelto.

