

Ejercicios Prácticos:

1. Clonando un Repositorio: Deben clonar un repositorio Git a su máquina local utilizando X comando. Pueden usar este <https://github.com/Matias-Villalobo/tutorial-de-git>.

```
User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP
$ git clone https://github.com/EnriqueKloosterman/fit-trabajo-practico-git.git
Cloning into 'fit-trabajo-practico-git'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP
$ |
```

2. Creación de Ramas: Creen una nueva rama en su repositorio clonado y luego cambien a la rama recién clonada. se puede utilizar el o los comandos de su preferencia.

```
User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP/fit-trabajo-practico-git (master)
$ git branch rama_1

User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP/fit-trabajo-practico-git (master)
$ git checkout rama_1
Switched to branch 'rama_1'

User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP/fit-trabajo-practico-git (rama_1)
$ |
```

3. Agregar y Confirmar Cambios: Realicen cambios en uno o más archivos dentro de la nueva rama y confirmen utilizando X comando. Deben incluir un mensaje descriptivo en cada confirmación.

```
User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP/fit-trabajo-practico-git (rama_1)
$ git commit -m "sepracion de headers"
[rama_1 8040c74] sepracion de headers
1 file changed, 2 insertions(+), 1 deletion(-)
```

```

User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP/fit-trabajo-practico-git (rama_1)
$ git push origin rama_1
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), 356 bytes | 89.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'rama_1' on GitHub by visiting:
remote:   https://github.com/EnriqueKloosterman/fit-trabajo-practico-git/pull/new/rama_1
remote:
To https://github.com/EnriqueKloosterman/fit-trabajo-practico-git.git
 * [new branch]      rama_1 -> rama_1

```

4. Fusionar Ramas: Después de hacer cambios en la nueva rama, fusionen esa rama con la rama principal utilizando X comando

```

User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP/fit-trabajo-practico-git (rama_1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

User@LAPTOP-AF9DL59P MINGW64 ~/Documents/CEPIT/Tercer Cuatrimestre/FIP/fit-trabajo-practico-git (master)
$ git merge rama_1
Updating 7d62ca3..8040c74
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

```

5. Resolución de Conflictos: Introduzcan deliberadamente un conflicto de fusión en el repositorio y resuélvanlo utilizando el comando de su elección.

Ejercicios Teóricos:

1. Explicar el Propósito de Git: Describan en sus propias palabras para qué se utiliza Git y por qué es importante en el desarrollo de software.

Es control de versiones donde alojamos nuestros proyectos en la nube teniendo acceso desde cualquier lado. Permite a varias personas trabajar a la vez. Mantiene un historial de los cambios hechos, y en caso de romper algo siempre se puede volver atrás a un punto que funcione correctamente.

2. Diferencia entre Commit y Push: Expliquen la diferencia entre los comandos git commit y git push.

Commit: Registra los cambios en el historial creando un snapshot de los cambios realizados.

Push: Envía los cambios hechos del repositorio local al repositorio remoto.

3. Ramas en Git: Cuál es el propósito y la utilidad de las ramas en Git.

Las ramas sirven para que varias personas puedan trabajar en el proyecto sin pisarse entre ellas y sin miedo a romper el código. Y de haber un error en alguna rama este no afecta al resto del proyecto.

4. Conflictos de Fusión: Expliquen qué son los conflictos de fusión en Git y cómo se pueden resolver.

Los conflictos ocurren cuando 2 o mas ramas trabajaron en la misma línea de un archivo. Se pueden resolver de forma manual decidiendo cual de los cambios mantener.

5. Flujo de Trabajo Básico de Git: Describan el flujo de trabajo básico de Git, desde la creación de un repositorio hasta la confirmación de cambios y la fusión de ramas.

- Crear el repo.

- Trabajar en el.

- Agregar cambios.

- Confirmar los cambios

- Crear y trabajar en ramas

- Fusionar las ramas.

- Resolver los conflictos

- Enviar los datos.

FECHA LÍMITE DE ENTREGA: 06/09/2023 A LAS 19 HRS