

TP2 – BIANCO NICOLAS

Actividades:

- **¿Qué es GitHub?**

Es una plataforma de desarrollo colaborativo basada en Git, que permite alojar, gestionar y compartir código de manera eficiente.

- **¿Cómo crear un repositorio en GitHub?**

Inicia sesión en GitHub.

Haz clic en el botón "+" en la esquina superior derecha y selecciona "New repository".

Especifica un nombre, descripción y visibilidad (público o privado). Haz clic en "Create repository".

- **¿Cómo crear una rama en Git?**

```
git branch nombre_rama
```

- **¿Cómo cambiar a una rama en Git?**

```
git checkout nombre_rama
```

- **¿Cómo fusionar ramas en Git?**

1 – Cambiar a la rama que se uno quiere fusionar

```
git checkout rama_principal
```

2 - Fusiona la otra rama:

```
git merge nombre_rama
```

- **¿Cómo crear un commit en Git?**

```
git add .
```

```
git commit -m "Mensaje"
```

- **¿Cómo enviar un commit a GitHub?**

```
git push origin nombre_rama
```

- **¿Qué es un repositorio remoto?**

Es un almacenamiento en línea de un proyecto Git, que permite compartir y sincronizar cambios entre diferentes colaboradores.

- **¿Cómo agregar un repositorio remoto a Git?**

```
git remote add origin <URL_DEL_REPOSITORIO>
```

- **¿Cómo empujar cambios a un repositorio remoto?**

```
git push origin nombre_rama
```

- **¿Cómo tirar de cambios de un repositorio remoto?**

```
git pull origin nombre_rama
```

- **¿Qué es un fork de repositorio?**

Un fork de un repositorio es una copia de un repositorio en tu cuenta de GitHub. Te permite modificar el código sin afectar el proyecto original. Los forks se utilizan comúnmente para colaborar en proyectos de código abierto o para experimentar con cambios antes de proponerlos al repositorio original mediante una solicitud de extracción (*pull request*).

- **¿Cómo crear un fork de un repositorio?**

Para crear un fork de un repositorio en GitHub, sigue estos pasos:

1. Inicia sesión en tu cuenta de GitHub.
2. Ve a la página del repositorio que desees forkar.
3. En la parte superior derecha, haz clic en el botón "Fork".
4. GitHub creará una copia del repositorio en tu cuenta.

Se puede clonar el fork al repositorio local:

```
git clone https://github.com/tu-usuario/nombre-del-repositorio.git
```

- **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?**

1 – Crear y subir cambios a una rama en el fork

```
git checkout -b mi-nueva-rama
```

```
git add .
```

```
git commit -m "Descripción de los cambios"
```

```
git push origin mi-nueva-rama
```

2 - Abre un Pull Request en GitHub

1. Ve a tu fork en GitHub.
2. Haz clic en el botón "Contribute" y selecciona "Open pull request".
3. Asegúrate de que la base (base repository) sea el repositorio original y la comparación (compare branch) sea tu rama con los cambios.
4. Escribe un título y una descripción explicando los cambios realizados.

5. Haz clic en "Create pull request".

Los mantenedores del repositorio original podrán revisar y decidir si aceptan tu contribución.

- **¿Cómo aceptar una solicitud de extracción?**

Navegar al repositorio donde se envió la solicitud de extracción, pestaña pull request, seleccionar a la que se debe aceptar ver cambios en files changed y fusionar la pull request, con el botón merge pull request, confirm merge.

- **¿Qué es un etiqueta en Git?**

Una etiqueta (tag) en Git es un marcador utilizado para identificar versiones específicas en la historia del repositorio, generalmente en momentos clave como el lanzamiento de una nueva versión de software.

Etiquetas ligeras (lightweight): simplemente apuntan a un commit específico.

Etiquetas anotadas (annotated): almacenan información adicional como el autor, fecha y mensaje.

- **¿Cómo crear una etiqueta en Git?**

```
git tag -a v1.0 -m "Versión 1.0"
```

- **¿Cómo enviar una etiqueta a GitHub?**

```
git push origin v1.0
```

- **¿Qué es un historial de Git?**

El historial de Git es un registro de todos los cambios realizados en un repositorio a lo largo del tiempo. Incluye información sobre cada commit, como:

- Identificador (hash) del commit
- Autor del cambio
- Fecha y hora del commit
- Mensaje del commit
- Archivos modificados

- **¿Cómo ver el historial de Git?**

```
git log
```

- **¿Cómo buscar en el historial de Git?**

Buscar por commits:

```
git log --grep="palabra clave"
```

Buscar commits por autor:

```
git log --author="Nombre del autor"
```

Buscar cambios en un archivo específico:

```
git log -- filename.txt
```

Buscar por fecha:

```
git log --since="2024-01-01"
```

Hasta una fecha:

```
git log --until="2024-07-01"
```

Buscar por contenido dentro de archivos:

```
git grep "texto buscado" $(git rev-list --all)
```

- **¿Cómo borrar el historial de Git?**

```
git log --until="2024-07-01"
```

- **¿Qué es un repositorio privado en GitHub?**

Un repositorio visible solo para el propietario y las personas invitadas. □

- **¿Cómo crear un repositorio privado en GitHub?**

Al crear un nuevo repositorio, selecciona la opción "Private".

- **¿Cómo invitar a alguien a un repositorio privado en GitHub?**

Configuración del repositorio > Settings, agregar un colaborador "Collaborators" en Manage Access, add people, se ingresa usuario de git o correo de la persona a invitar.

- **¿Qué es un repositorio público en GitHub?**

Un repositorio público en GitHub es un repositorio accesible para cualquier persona en Internet. Cualquier usuario puede verlo, clonarlo y descargar su contenido, pero solo aquellos con permisos pueden hacer cambios.

- **¿Cómo crear un repositorio público en GitHub?**

Modificar acceso y seleccionar "Public"

- **¿Cómo compartir un repositorio público en GitHub?**

Con la URL del repositorio: <https://github.com/usuario/repositorio>

Ejercicio 1:

```
Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1
$ git init
Initialized empty Git repository in C:/Users/Nico/Desktop/GIT/E0/Ejercicio 1/.git/

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$ git add .

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$ git commit -m "Proyecto 1"
[master (root-commit) 9f5ea6f] Proyecto 1
1 file changed, 11 insertions(+)
create mode 100644 file.html

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$ git remote add origin https://github.com/nicoxango994/Ejercicio1.git

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 359 bytes | 359.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nicoxango994/Ejercicio1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$ git add .

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$ git commit -m "TITULO DE PAGINA"
[master 644a6d8] TITULO DE PAGINA
1 file changed, 1 insertion(+), 1 deletion(-)

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 297 bytes | 297.00 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.
To https://github.com/nicoxango994/Ejercicio1.git
 9f5ea6f..644a6d8 master -> master
branch 'master' set up to track 'origin/master'.

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E0/Ejercicio 1 (master)
$
```

URL: <https://github.com/nicoxango994/Ejercicio1>

Ejercicio 2:

```
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.

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E1
$ cd conflict-exercise

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E1/conflict-exercise (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E1/conflict-exercise (feature-branch)
$ git add .

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E1/conflict-exercise (feature-branch)
$ git status
On branch feature-branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

Nico@DESKTOP-RL5IK4K MINGW64 ~/Desktop/GIT/E1/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch de92565] Added a line in feature-branch
1 file changed, 2 insertions(+), 1 deletion(-)
```

URL: <https://github.com/nicoxango994/conflict-exercise>