

Actividad1

¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo que utiliza Git para la gestión de versiones. Permite almacenar proyectos de software en repositorios y facilita el trabajo en equipo, así como la colaboración mediante "pull requests".

¿Cómo crear un repositorio en GitHub?

Accede a tu cuenta en GitHub.

Haz clic en el botón "New" o "Nuevo repositorio".

Asigna un nombre y una descripción (opcional).

Decide si el repositorio será público o privado.

Selecciona "Initialize this repository with a README" si quieres agregar un archivo README.

Haz clic en "Create repository"

¿Cómo crear una rama en Git?

git branch nombre-de-la-rama

¿Cómo cambiar a una rama en Git?

git checkout nombre-de-la-rama

¿Cómo fusionar ramas en Git?

Desde la rama donde se desea hacer la fusión:

git checkout rama-principal

Se fusiona la rama :

git merge nombre-de-la-rama

¿Cómo crear un commit en Git?

Después de hacer cambios, se agregan los archivos al área de preparación:

git add .

Después se crea el commit:

git commit -m "Mensaje del commit"

¿Cómo enviar un commit a GitHub?

Después de hacer el commit, sube los cambios a GitHub con:

git push origin nombre-de-la-rama

¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu repositorio almacenada en un servidor, como GitHub, GitLab o Bitbucket. Te permite compartir y colaborar con otros usuarios.

¿Cómo agregar un repositorio remoto a Git?

git remote add origin https://github.com/usuario/nombre-repositorio.git

¿Cómo empujar cambios a un repositorio remoto?

git push origin nombre-de-la-rama

¿Cómo tirar de cambios de un repositorio remoto?

git pull origin nombre-de-la-rama

¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio que te permite hacer cambios sin afectar al repositorio original. Es útil para contribuir a proyectos de otras personas.

¿Qué es un fork de repositorio?

Un fork es una copia de un repositorio que te permite hacer cambios sin afectar al repositorio original. Es útil para contribuir a proyectos de otras personas.

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

Después de realizar un fork y cambios en tu copia del repositorio, ve a la página de tu repositorio en GitHub y haz clic en "**New Pull Request**". Luego, selecciona la rama que deseas fusionar con el repositorio original.

¿Cómo aceptar una solicitud de extracción?

En GitHub, ve a la sección de **Pull Requests** del repositorio. Haz clic en la solicitud y selecciona "**Merge pull request**" para aceptarla.

¿Qué es una etiqueta en Git?

Una etiqueta es una referencia a un commit específico. Es útil para marcar versiones del proyecto.

¿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 commits realizados en un repositorio, mostrando qué cambios se hicieron, cuándo y por quién.

¿Cómo ver el historial de Git?

git log

¿Cómo buscar en el historial de Git?

git log --grep="palabra-clave"

¿Cómo borrar el historial de Git?

git reset --hard HEAD~n

n es el número de commits a eliminar.

¿Qué es un repositorio privado en GitHub?

Un repositorio privado es un repositorio al que solo pueden acceder personas específicas que hayas invitado. No es visible para el público.

¿Cómo crear un repositorio privado en GitHub?

Cuando creas un repositorio en GitHub, selecciona "**Private**" en lugar de "**Public**".

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

Ve a la pestaña "**Settings**" de tu repositorio y selecciona "**Collaborators**". Agrega el nombre de usuario de la persona que deseas invitar.

¿Qué es un repositorio público en GitHub?

Un repositorio público es un repositorio visible para cualquier persona. Cualquiera puede ver, clonar y bifurcar el repositorio.

¿Cómo crear un repositorio público en GitHub?

Al crear un repositorio en GitHub, selecciona la opción "**Public**".

¿Cómo compartir un repositorio público en GitHub?

Compartiendo la URL de tu repositorio

¿Cómo clonar un repositorio desde GitHub?

git clone URL del repositorio

¿Cómo agregar un archivo a un repositorio de Git?

Después de crear o modificar un archivo, usar los siguientes comandos:

git add nombre-del-archivo

git commit -m "Mensaje descriptivo"
git push origin nombre-de-la-rama

¿Cómo crear una nueva rama en Git y cambiar a ella?

git checkout -b nueva-rama

¿Cómo hacer un commit en Git y enviarlo a GitHub?

git add .
git commit -m "Descripción del commit"
git push origin nombre-de-la-rama

¿Cómo eliminar un repositorio en GitHub?

Entra al repositorio en GitHub, ve a **Settings**, desplázate hacia abajo y en la **Danger Zone**, haz clic en **Delete this repository**.

Actividad2

Crear un repositorio.

<https://github.com/anahi651/tp2actividad2>

- Agregando un Archivo

```
MINGW64:/c/Users/Ani/Desktop/progracionTp2Act2

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$ git init
Initialized empty Git repository in C:/Users/Ani/Desktop/progracionTp2Act2/.git/

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$ git add .

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$ git commit -m "Agregando mi-archivo.txt"
[master (root-commit) e89d11f] Agregando mi-archivo.txt
1 file changed, 20 insertions(+)
create mode 100644 mi-archivo.txt

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$ git remote add origin https://github.com/anahi651/tp2actividad2.git


Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 469 bytes | 469.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/anahi651/tp2actividad2.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$
```

Creando Branchs

```
Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$ git branch nuevarama

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (master)
$ git checkout nuevarama
Switched to branch 'nuevarama'

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (nuevarama)
$ touch archivo-en-nueva-rama.txt

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (nuevarama)
$ git add .

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (nuevarama)
$ git commit -m "se crea archivo"
[nuevarama 87eca56] se crea archivo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 archivo-en-nueva-rama.txt

Ani@Anahi MINGW64 ~/Desktop/progracionTp2Act2 (nuevarama)
$ git push origin nuevarama
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'nuevarama' on GitHub by visiting:
remote:   https://github.com/anahi651/tp2actividad2/pull/new/nuevarama
remote:
To https://github.com/anahi651/tp2actividad2.git
* [new branch]      nuevarama -> nuevarama
```

Actividad3

Paso 1: Crear un repositorio en GitHub

<https://github.com/anahi651/conflict-exercise>.

```
Ani@Anahi MINGW64 ~ (master)
$ git clone https://github.com/anahi651/conflict-exercise
Cloning into 'conflict-exercise'...
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.

Ani@Anahi MINGW64 ~ (master)
$ cd conflict-exercise
```

```
Ani@Anahi MINGW64 ~/conflict-exercise (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

Ani@Anahi MINGW64 ~/conflict-exercise (feature-branch)
$ git add README.md

Ani@Anahi MINGW64 ~/conflict-exercise (feature-branch)
$ git commit -m "Added a line in feature-branch"
[feature-branch 6c0d10f] Added a line in feature-branch
1 file changed, 1 insertion(+)

Ani@Anahi MINGW64 ~/conflict-exercise (feature-branch)
$ |
```

```
Ani@Anahi MINGW64 ~/conflict-exercise (main/MERGING)
$ git add README.md

Ani@Anahi MINGW64 ~/conflict-exercise (main/MERGING)
$ git commit -m "Resolved merge conflict"
[main ec555de] Resolved merge conflict

Ani@Anahi MINGW64 ~/conflict-exercise (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 821 bytes | 273.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/anahi651/conflict-exercise
   2bf9859..ec555de  main -> main

Ani@Anahi MINGW64 ~/conflict-exercise (main)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:   https://github.com/anahi651/conflict-exercise/pull/new/feature-bran
ch
remote:
To https://github.com/anahi651/conflict-exercise
 * [new branch]   feature-branch -> feature-branch

Ani@Anahi MINGW64 ~/conflict-exercise (main)
```