Git avanzado

| **Summary** | En este codelab aprenderás sobre los conceptos de git, Github, como utilizarlos y sus diferencias, a través de ejercicios prácticos mediante el uso adecuado de ramas. |
| --- | --- |
| **URL** | cp5 |
| **Category** | Web |
| **Environment** | web |
| **Status** | Published |
| **Feedback Link** |  |
| **Author** | Marlene Aguilar |

[Introducción](#_wsuihf2e4v1t)

[Comandos Importantes](#_z8hukrx14ix1)

[Ramas(Realizar el ejemplo en la práctica)](#_as2xgwuv9vk)

[Teoria](#_jfpmp3ghgacs)

[Ejemplo a realizar:](#_38gcn1ef6k2b)

[Estados en Git](#_xb5cp5n96zkk)

[El flujo de trabajo básico de Git](#_bw12sikg200i)

[Ejercicio evaluado](#_j83cbau04aqs)

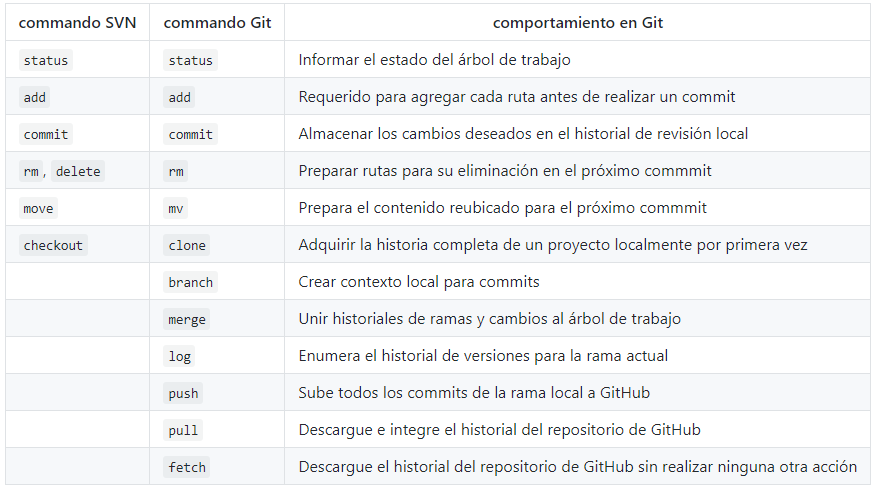
# 

# 

# Introducción

En esta guía aprenderás sobre qué es Git y Github, mediante la implementación de ramas las cuales son una forma de mantener varias líneas de desarrollo separadas y estas nos pueden ayudar a evitar que los cambios no deseados afecten el código principal.

## Comandos Importantes



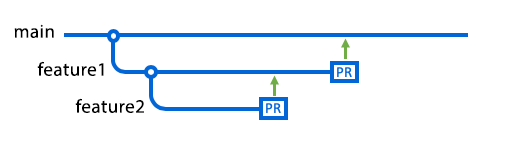
Leer más: [comandos-git.pdf](https://drive.google.com/file/d/1sAYpx-N9RcNDpeurbF8m7gggJsdgt_pX/view?usp=sharing)

# Ramas(Realizar el ejemplo en la práctica)

## Teoria

La rama por defecto en git es la rama master (main), las ramas permiten trabajar de forma separada las secciones de un proyecto sin que la rama principal se vea afectada, las ramas se crean en paralelo, se puede cambiar de una rama a otra cuando se considera necesario.

En la siguiente imagen se muestran **3 ramas**.



**Comandos**

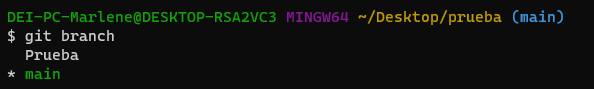
Crear una nueva rama:

* git branch＜new-branch＞



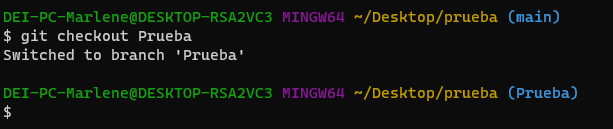
Mostrar las ramas (la rama en uso se resalta)

* git branch



Cambiarse de rama

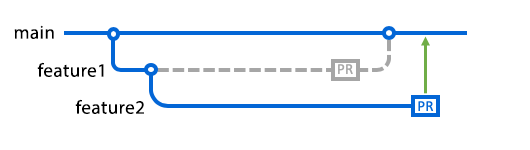
* git checkout＜new-branch＞



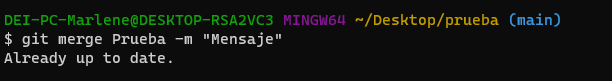
**Fusión**

Esta acción permite unir los cambios realizados en las ramas con la rama principal. Para lo cual se debe ubicar en la rama base y aplicar el comando correspondiente.

En la imagen se muestra los cambios realizados en la rama feature1 se han fusionado con la rama principal (main).



* git merge



| **Nota:** En la fusión pueden suceder conflictos los cuales se deben resolver de forma manual en muchos de los casos. |
| --- |

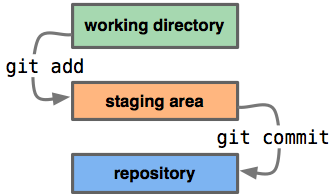
## Ejemplo a realizar:

Solicitar al usuario dos números enteros, luego crear las siguientes funcionalidades en ramas diferentes:

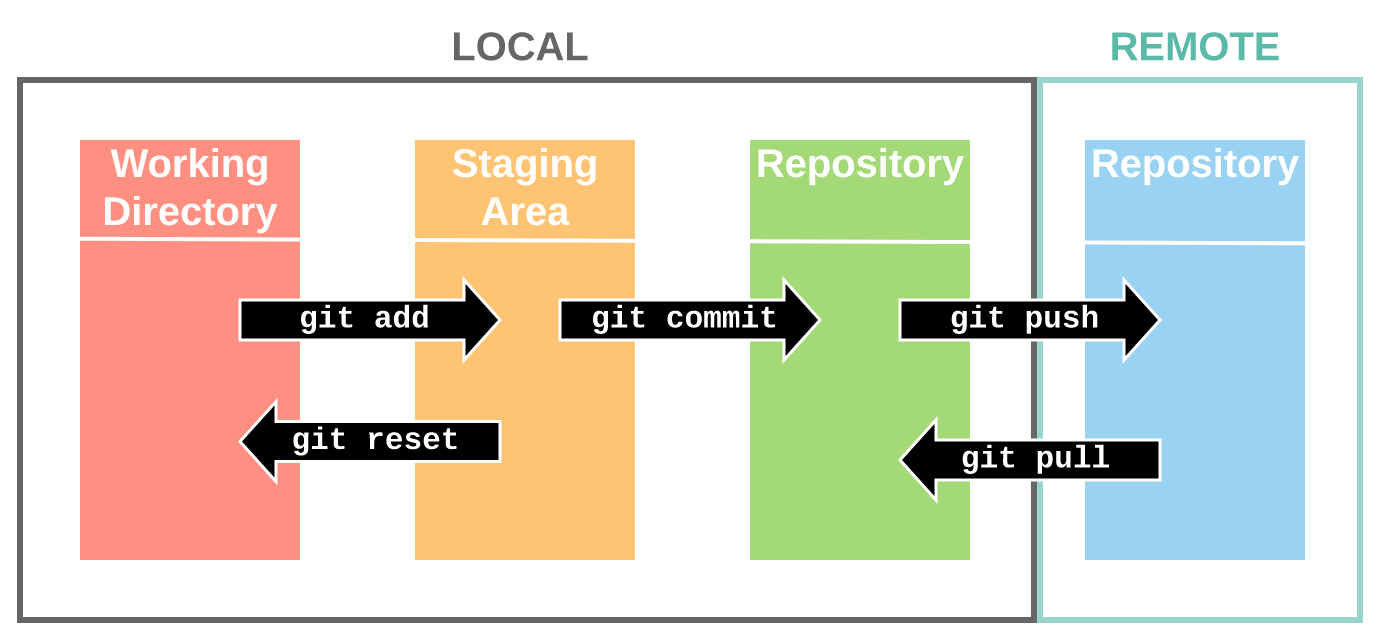
* Crea una función que tome dos números y devuelva su suma.
* Crear una función que tome dos números enteros y devuelva su resta.

## Estados en Git

* Confirmado (committed): la información está almacenada en la base de datos local.
* Modificado (modified): un archivo ha sido modificado, pero se han confirmado a la base de datos.
* Preparado (staged): se ha marcado un archivo modificado en su versión actual para enviarlo a la próxima confirmación.

****

## El flujo de trabajo básico de Git

****

**Los estados y el flujo de trabajo es el mismo en cualquier rama dentro de nuestro repositorio.**

Leer más: [Estados Git](https://moisesdelacruz.medium.com/tutorial-b%C3%A1sico-de-git-y-github-42e46ff41194)

Ver video**:** [Git & GitHub: Git Status](https://www.youtube.com/watch?v=SxmveNrZb5k)

# 

# Ejercicio evaluado

1. Formar grupos de 3 personas
2. Un integrante debe acceder a la tarea que está en Moodle llamada cp5 y deberá crear un grupo en github classroom. Los demás integrantes deberán unirse al grupo creado
3. Todos los integrantes deberán clonar el repositorio
4. Crea una rama(Por estudiante) en el repositorio clonado y sube los cambios a esa rama
5. Realizar el ejercicio que asigne el instructor/a
6. Sube los cambios al repositorio utiliza los comandos en el orden correcto (git add, git commit, git push), mínimo de commit 3
7. Realiza la fusión de la rama con el main
8. Puedes verificar en tu repositorio de GitHub los cambios que realizaste al código.
9. Borrar las credenciales de windows
10. Borrar de la computadora el o los programas creados durante la práctica