

# Git

Pedro O. Pérez M., PhD.

Herramientas computacionales: el arte de la analítica  
Tecnológico de Monterrey

*pperezm@tec.mx*

03-2024

- 1 ¿Qué es Git?
  - Terminología
  - Características
  
- 2 Operaciones sobre repositorios
  - Crear un repositorio
  - Clonar un repositorio
  - Subir archivos al repositorio
  - Administrar ramas
  - Fusionar ramas
  
- 3 ¿Qué es un conflicto?

# ¿Qué es Git?

- Git es un sistema de control de versiones que permite registrar los cambios realizados a archivos de computadoras con el fin de coordinar los avances de un equipo de trabajo.



# Terminología

- **Repositorio:** Carpeta del proyecto.
- **Commit:** Cambio realizado a un proyecto.
- **Rama:** Copia aislada del proyecto en que una persona puede realizar su respectivo trabajo.

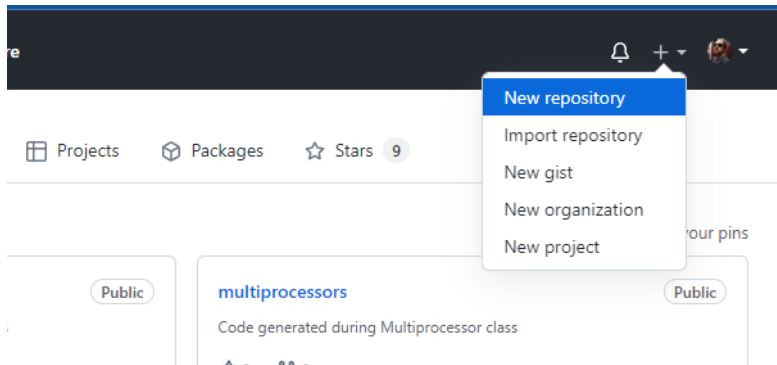


# Características

- **Distribuido:** No hay una conexión central a Internet o a un servidor.
- **Ramas:** Las diversificaciones del proyecto se pueden unir después.
- **Integridad de datos:** Asegura que cada persona tenga los mismos datos.



# Crear un repositorio



# Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

## Repository template

Start your repository with a template repository's contents.

No template ▾

Owner \*

 Manchas2k4 ▾

Repository name \*

/

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

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:

3

Skip this step if you're importing an existing repository.



Add a README file

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



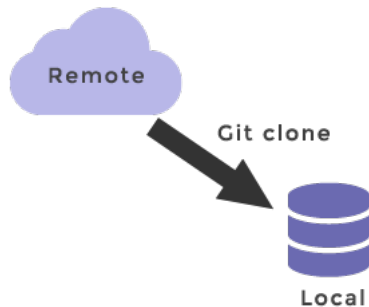
Add .gitignore

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

# Clonar un repositorio

Clona el repositorio en tu computadora.

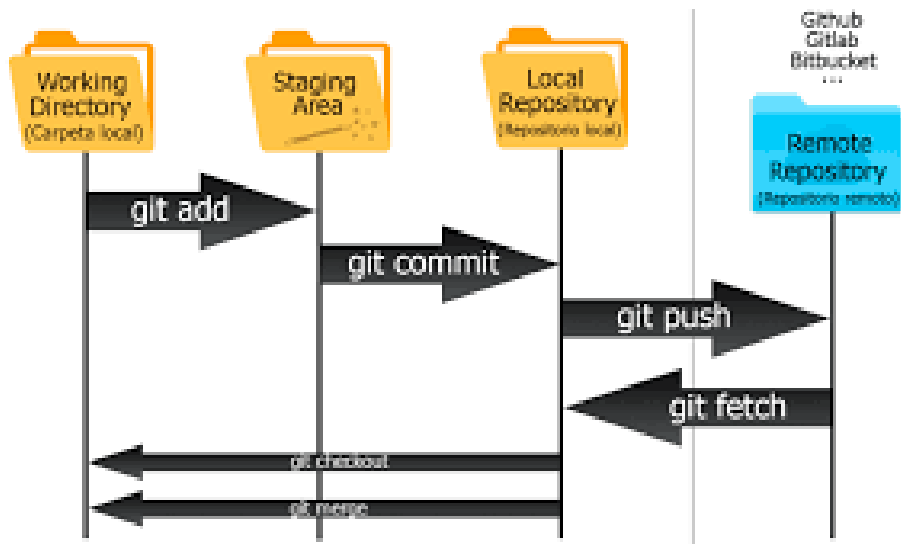
```
git clone url_del_repositorio
```





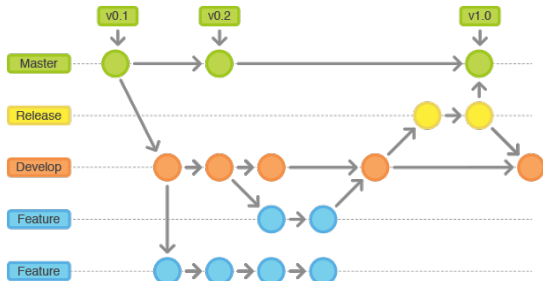
# Subir archivos al repositorio

- ① `git add .` - Sube los cambios realizados a los archivos a una zona de pruebas para que Git empiece a “rastrearlos”.
- ② `git commit -m “Mensaje”` - Confirma que los cambios realizados son definitivos, subiendo el contenido al repositorio local. El mensaje debe explicar a detalle qué cambio se realizado.
- ③ `git origin [rama]` - Sube el contenido al repositorio remoto.



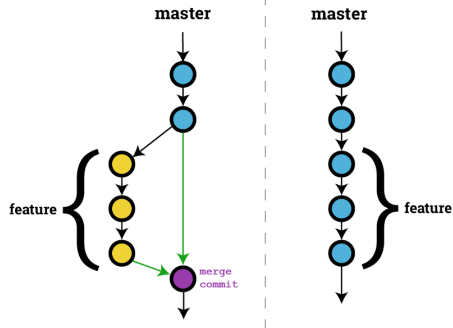
# Administrar ramas

- 1 `git checkout`: Cambia de rama.
- 2 `git checkout -b`: Crea una nueva rama y se cambia a ella.
- 3 `git branch`: Enlista todas las ramas del repositorio.
- 4 `git branch -d`: Elimina la rama actual.
- 5 `git status`: Identifica en cuál rama estamos actualmente.



# Fusionar ramas

- 1 `git pull`: Carga el contenido de una rama en el repositorio.
- 2 `git merge`: Fusiona el contenido de dos ramas.



## ¿Qué es un conflicto?

- Es una inconsistencia entre los archivos de dos ramas de un repositorio. Se da cuando dos personas cambiaron la misma porción de un código.

# ¿Qué hacer ante un conflicto?

- Hay varios entornos de desarrollo (IDE) y editores de texto que marcan las partes de código que están en conflicto. Estos son algunos de ellos:



(a) VS Code



(b) IntelliJ IDEA



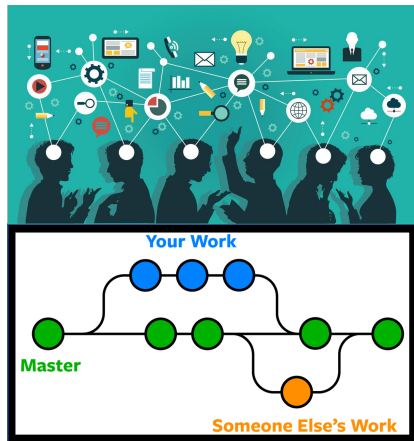
(c) Sublime Text 3



(d) Atom

# ¿Cómo evitar un conflicto ?

- 1 La mejor forma de evitar conflictos al fusionar ramas es mediante la integración continua de avances. Esto significa subir de forma constante tus avances al repositorio para que las diferencias de código entre tu rama y las de los demás no sea tan grande. Esto requiere que haya comunicación constantes de tu equipo para informar los cambios que se hacen al código.



- Existen varias empresas que ofrecen espacios para repositorios remotos en la nube. Algunas de las más importantes son:



(a) Github



(b) Bitbucket



(c) Gitlab



Comando	Descripción
<code>git clone url</code>	Clonar un repositorio.
<code>git branch</code>	Enlistar las ramas.
<code>git checkout -b 'destino' 'origen'</code>	Crear una nueva rama (destino) y cambiar.
<code>git checkout 'destino'</code>	Cambia de rama.
<code>git status</code>	Revisar en qué rama estamos.
<code>git add . (git add -A)</code>	Sube los cambios realizados a los archivos a una zona de pruebas para que Git empiece a "rastrearlos".
<code>git commit -m "Mensaje"</code>	Sube el contenido al repositorio local.
<code>git push origin dev</code>	Sube el contenido al repositorio local en la rama 'dev'.
<code>git merge 'dev'</code>	Combina los archivos en 'dev' con los archivos 'main'. Es importante, antes de hacer el 'merge', ingresar a la rama deseada.