

# Desarrollo Backend con NodeJS

Git y github

# Conceptos generales

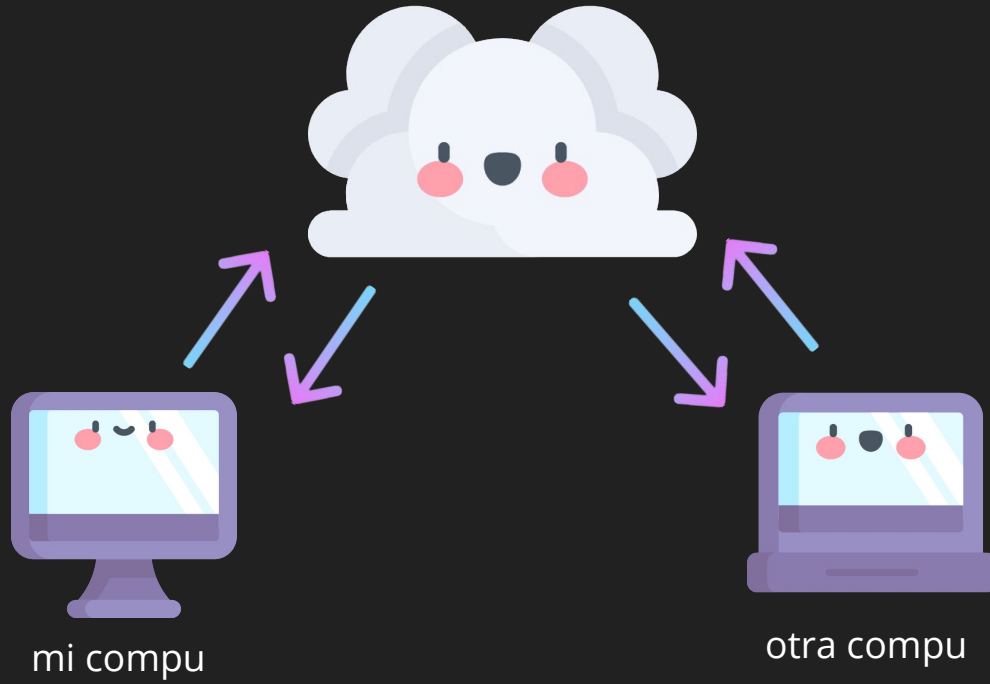
# Repositorio

El repositorio es el lugar en el que se almacenan los datos actualizados y el histórico de cambios. Como referencia podemos pensar en una carpeta de Google Drive

# Versión

Una versión es un estado determinado de el archivo que estamos revisando. Las versiones se identifican mediante un código de detección de modificaciones. A la última versión (la más actual) se la suele identificar de forma especial con el nombre de HEAD

repositorio



# ¿Qué vamos a utilizar?



Git, herramienta de versionado, se instala en nuestra compu y se utiliza a través de la terminal

<https://git-scm.com/>



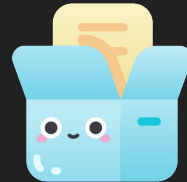
Github, plataforma para compartir repositorios

<https://github.com/>

## Comandos basicos de la terminal

- `cd NOMBRE_CARPETA`
  - Sirve para movernos dentro de una carpeta
- `cd ..`
  - Nos lleva un nivel hacia arriba en nuestra estructura de carpetas
- `ls`
  - Lista archivos dentro de una carpeta
- `pwd`
  - Muestra la dirección completa donde estamos
- `mkdir`
  - Crea un directorio con el nombre especificado

## Comandos básicos de git



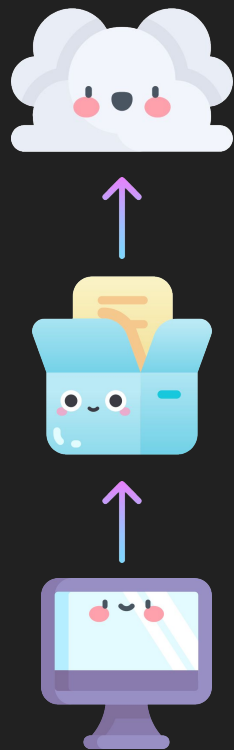
- `git clone`
  - clona un repositorio remoto a nuestra computadora.



## Comandos básicos de git

### Para enviar archivos al remoto:

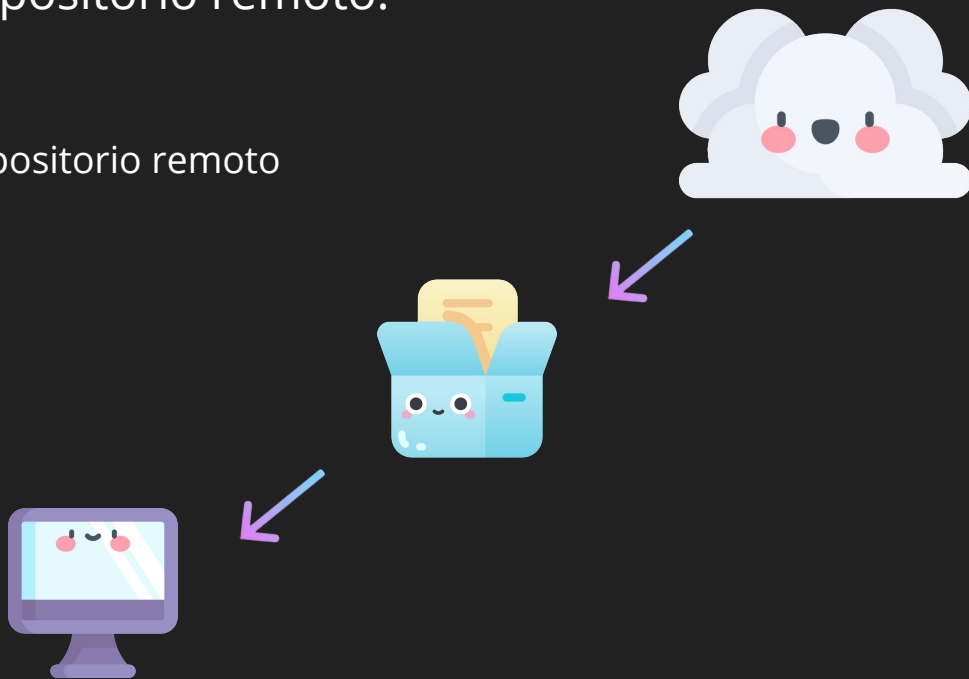
- `git status`
  - me dice si tengo cambios para subir
- `git add`
  - agrega mis cambios para que luego subirlos a nuestro repositorio remoto
- `git commit`
  - genera una nueva versión con nuestro código actualizado
- `git push`
  - sube los cambios desde nuestro repositorio local al repositorio remoto
  - <https://teloexplicocongatitos.com/poster/tlecg04>



## Comandos básicos de git

Para recibir información del repositorio remoto:

- `git pull`
  - trae la última versión del repositorio remoto



# Git pull

Imaginemos esta situación:

Nuestro local contiene archivo1.js al que le realizamos modificaciones y en el remoto tenemos cambios en archivo2.js.

Al hacer **pull** git trae los cambios sobre el archivo 'archivo2.js', como los cambios locales fueron sobre archivo1.js la actualización se realiza automáticamente

# Merge

Ahora otra persona envió sus cambios sobre el archivo1.js al remoto, **git pull** trae los cambios locales y debe realizar un **merge**

- Merge automático
  - Git tiene la capacidad de ordenar los cambios de forma automática siempre y cuando no pisen la misma línea de código
- Merge manual
  - Git no tiene la capacidad de ordenar los cambios de forma automática cuando son realizados sobre la misma línea. Esto se denomina conflicto y git no pedirá que lo resolvamos, es decir que tendremos que revisar el código a mano y decidir que cambios mantener

# Merge

- Merge automático
  - Git tiene la capacidad de ordenar los cambios de forma automática siempre y cuando no pisen la misma línea de código



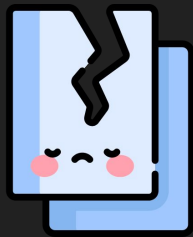
archivo1.js local  
cambio en línea 38



archivo1.js remoto  
cambio en línea 38

# Merge

- Merge manual



archivo1.js local  
cambio en línea 38



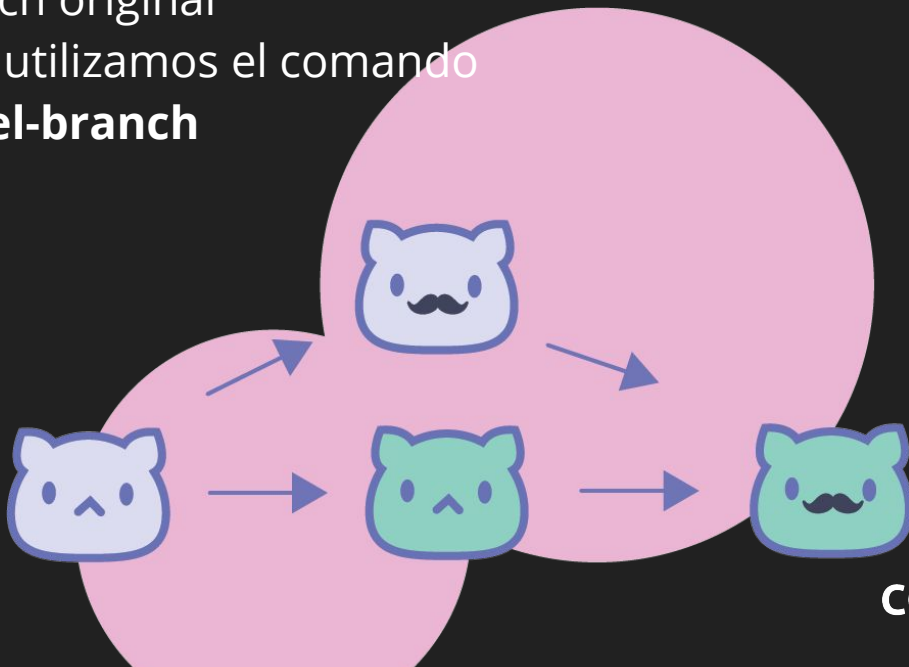
archivo1.js remoto  
cambio en línea 38

# Branches

- 'Branch' en español rama es una línea temporal paralela con su propio histórico de cambios donde podemos trabajar sobre otra versión de nuestro código sin impactar la branch original
  - para crear una branch utilizamos el comando **git branch nombre-del-branch**

con-bigotes

main (o master)



# Pull request

El pull request es la acción de informar a una branch base que queremos mergear nuestros cambios de otra branch de vuelta a su histórico

