



Git: Es un sistema de control de versiones

Control de versiones: Sistema que registra los cambios en un archivo o conjunto de archivos a lo largo del tiempo para que se puedan recuperar versiones específicas más adelante.

Git Hub: Es uno de los servicios de repositorio en nube más conocidos en el mundo, permite el uso de repositorios públicos de forma gratuita, así como de una cantidad limitada de repositorios privados (ampliable mediante suscripción), algo muy utilizado por muchas empresas para el control de versiones de los proyectos.



`git init`

Crea un nuevo repositorio de Git. Puede utilizarse para convertir un proyecto existente y sin versión en un repositorio de Git, o para inicializar un nuevo repositorio vacío. La mayoría de los demás comandos de Git no se encuentran disponibles fuera de un repositorio inicializado, por lo que este suele ser el primer comando que se ejecuta en un proyecto nuevo.

Al ejecutar `git init`, se crea un subdirectorio de `.git` en el directorio de trabajo actual, que contiene todos los metadatos de Git necesarios para el nuevo repositorio. Estos metadatos incluyen subdirectorios de objetos, referencias y archivos de plantilla.

Academia Java 2023

CARINA VÁSQUEZ PÉREZ

ESTADOS DE LOS ARCHIVOS AL TRABAJAR CON GIT

git status

Muestra el estado del directorio de trabajo y el área de ensayo. Permite ver qué cambios se han realizado, cuáles no y qué archivos no están siendo rastreados por Git.

Esto se usa para explorar la historia de un código específico y responder preguntas sobre qué, cómo y por qué se agregó el código a un repositorio.

git blame

git add

Agrega el contenido del archivo al índice

git add .

Añade todos los archivos

git add nomb_ar

Añade un archivo en específico denotando la ruta de éste.

Quita archivos del historial o los devuelve a su estado anterior

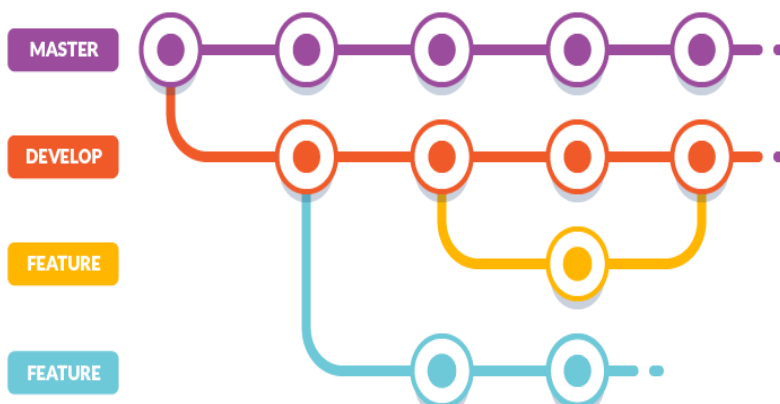
git reset HEAD

git rm

Eliminará el directorio de destino y todo el contenido de dicho directorio.

TRABAJANDO CON RAMAS

Un rama (branch) en Git es una forma de separar nuestro trabajo del trabajo ya existente en el repositorio.



Queremos trabajar en algo que no es definitivo o estable y no lo queremos subir a master por si rompemos algo que funciona. En este caso crearíamos una rama aparte en la que trabajar, y cuando nuestro trabajo esté listo, la fusionaríamos con master

\$ git branch <nombre_rama>

Para crear una rama nueva

Podremos ver las ramas que están creadas mediante el comando `git branch`, sin indicar un nombre. El asterisco señala nuestra rama actual

git branch

```
HP@DESKTOP-3UQQHPL MINGW64 ~/Documents/XideralJava/Ejercicios (master)
$ git branch
* master
  tmp
```

\$ git checkout <nombre_rama>

Para cambiar a la rama que queremos especificamos su nombre

Crear la rama y cambiar de ella en un solo comando

\$ git checkout -b <nombre_rama>

\$ git push origin <rama>

Subir ramas al repositorio remoto

Ver la lista de ramas

git branch or git branch --list

git branch -d <branch-name>

Para borrar ramas.

Compartir y Actualizar Proyectos

git fetch

Comunica con un repositorio remoto y obtiene toda la información que se encuentra en ese repositorio que no está en el tuyo actual y la almacena en tu base de datos local.

Git descargará desde el repositorio remoto especificado y a continuación, de forma inmediata intentará combinarlo en la rama en la que te encuentres.

git pull

git commit -m "commit message"

Una vez que llegamos a cierto punto en el desarrollo, queremos guardar nuestros cambios. También necesitamos escribir un mensaje corto para explicar lo que hemos desarrollado o cambiado en el código fuente.

git push

Se utiliza para comunicar con otro repositorio, calcular lo que tu base de datos local tiene que la remota no tiene, y luego subir (push) la diferencia al otro repositorio.

Para ver los repositorios remotos que tienes configurados.

Mostrará los nombres de cada uno de los remotos que tiene especificados. Si ha clonado su repositorio, debería ver al menos **origin** (origen, en inglés) - este es el nombre que por defecto Git le da al servidor del que ha clonado:

git remote

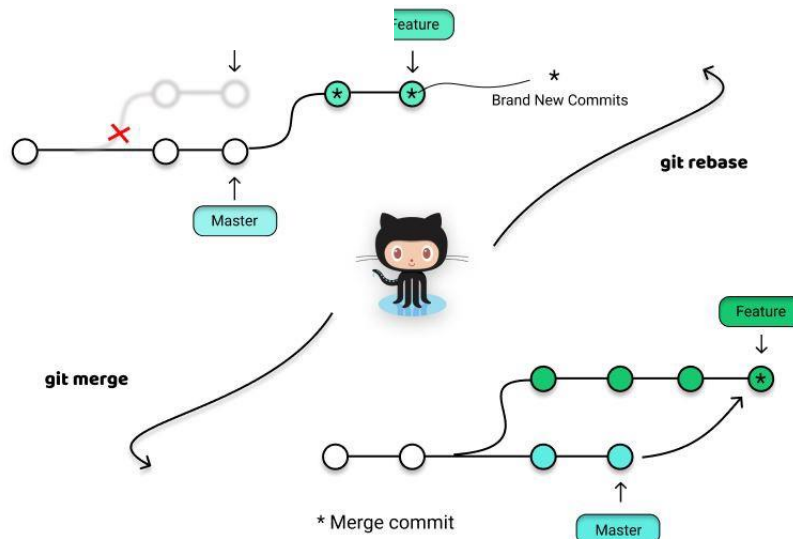
\$ git push origin <rama>

Subir ramas al repositorio remoto

Fusionar ramas

Integra su rama de características (feature branch) con todas sus confirmaciones en la rama dev (o master). Es importante recordar que primero debe estar en la rama específica que desea fusionar con su rama de características.

git merge <nombre_rama>



RESOLVIENDO CONFLICTOS



Si las dos ramas que intenta fusionar cambiaron la misma parte del mismo archivo, Git no podrá determinar qué versión usar. Cuando ocurre tal situación, se detiene justo antes de la confirmación de fusión para que pueda resolver los conflictos manualmente.

Cuando encuentra un conflicto de combinación, ejecutar el `git status` comando le muestra qué archivos deben resolverse.

```
On branch main
Unmerged paths:
(use "git add/rm ..." as appropriate to mark resolution)
both modified: hello.py
```

Ambas ramas
modificarán la
misma sección
de hello.py

Marcadores visuales

```
<<<<<<<
===
>>>>>>>
```

Es útil buscar estos indicadores en un proyecto durante una fusión para encontrar dónde se deben resolver los conflictos.

Generalmente, el contenido antes del `===` marcador es la rama receptora y la parte posterior es la rama de fusión.

Una vez que haya identificado las secciones en conflicto, puede entrar y arreglar la fusión a su gusto. Cuando esté listo para finalizar la combinación, todo lo que tiene que hacer es ejecutar `git add` los archivos en conflicto para decirle a Git que están resueltos. Luego, ejecuta un normal `git commit` para generar la confirmación de fusión.