

Capacitacion Sistema API Rest

Git & GitLab

Descarga GIT

- Link de descarga de git: [Home Page](#)

¿Que es un Repositorio de Git?

Un repositorio de Git es un almacenamiento virtual de tu proyecto. Te permite guardar versiones del código a las que puedes acceder cuando lo necesitas.

Inicio de un nuevo repositorio: `git init`

Para crear un nuevo repositorio, usa el comando `git init`.

`git init` es un comando que se utiliza una sola vez durante la configuración inicial de un repositorio nuevo. Al ejecutar este comando, se creará un nuevo subdirectorio `.git` en tu directorio de trabajo actual. También se creará una nueva rama maestra.

- Apuntar `git init` a un directorio de proyecto existente hará que se ejecute la misma configuración de inicio mencionada arriba, pero en el ámbito de ese directorio.

```
git init
```

Clonación de un repositorio existente: `git clone`

Si un repositorio ya se ha configurado en un repositorio central, el comando de clonación es la manera más común de obtener una copia de desarrollo local. Igual que `git init`, la clonación suele ser una operación única. Una vez que un desarrollador ha obtenido una copia de trabajo, todas las operaciones de control de versiones se administran por medio de su repositorio local.

- Ejecutar el siguiente comando donde `<url>` es la ruta del repositorio.

```
git clone <url>
```

Guardar cambios en el repositorio: `git add` y `git commit`

Ahora que has iniciado o clonado un repositorio, puedes realizar commits en la versión del archivo. Para el siguiente ejemplo asumiremos que has configurado un proyecto en `/path/to/project`. Los pasos son los siguientes:

- Cambia el directorio a `/path/to/project`
- Crea un archivo nuevo `CommitTest.txt` con el contenido "test content for git tutorial"
- Ejecuta el comando `git add` para añadir `CommitTest.txt` al área de preparación del repositorio

- Crea un commit nuevo con un mensaje que describa qué trabajo se ha hecho en el commit.

CommitTest.txt

```
git add CommitTest.txt
```

```
# git add . o --all Agrega los archivos y directorios con contenido.
```

```
git commit -m "added CommitTest.txt to the repo"
```

Sincronizacion de Git



Características principales:

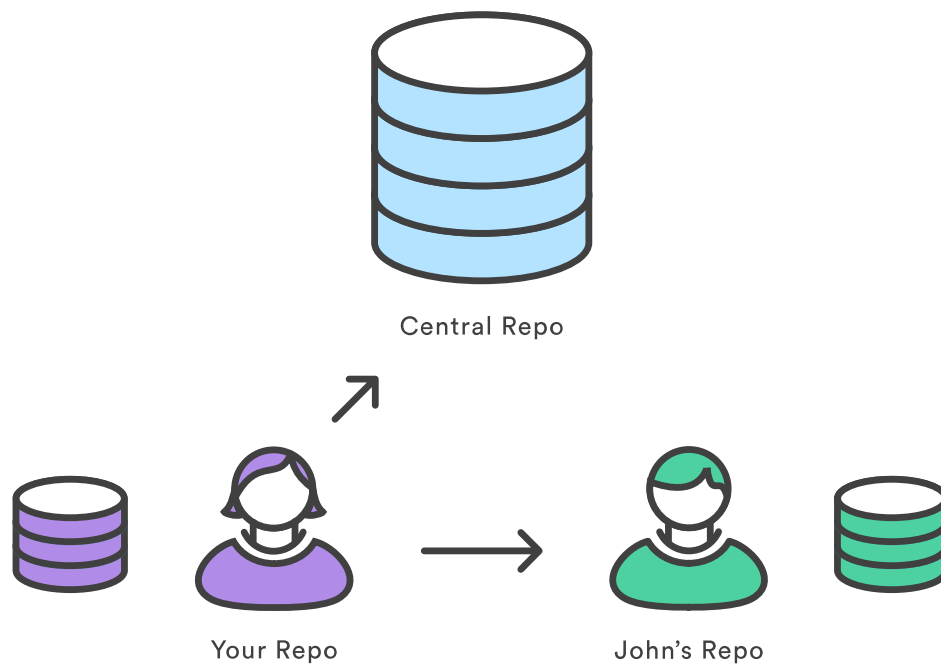
- git remote
- git fetch
- git push
- git pull

El comando `git remote` es una parte de un sistema más amplio que se encarga de sincronizar los cambios. Los registros inscritos mediante el comando `git remote` se utilizan junto con los comandos `git fetch`, `git push` y `git pull`. Todos estos comandos tienen sus propias responsabilidades de sincronización, que pueden consultarse en los enlaces correspondientes.

Git remote

El comando `git remote` te permite crear, ver y eliminar conexiones a otros repositorios. Las conexiones remotas se asemejan más a marcadores que a enlaces directos con otros repositorios. En lugar de brindar acceso en tiempo real a otro repositorio, funcionan como prácticos nombres que pueden emplearse para hacer referencia a una URL no tan sencilla.

Por ejemplo, en el siguiente diagrama pueden verse dos conexiones remotas desde tu repositorio hasta el repositorio central y el repositorio de otro desarrollador. En lugar de hacer referencia a ellas mediante sus URL completas, puedes pasar los atajos del origen y John a otros comandos de Git.



Enumera las conexiones remotas que tienes con otros repositorios.

```
git remote
```

Lo mismo que el comando anterior, pero incluye la URL de cada conexión.

```
git remote -v
```

Uso de git push

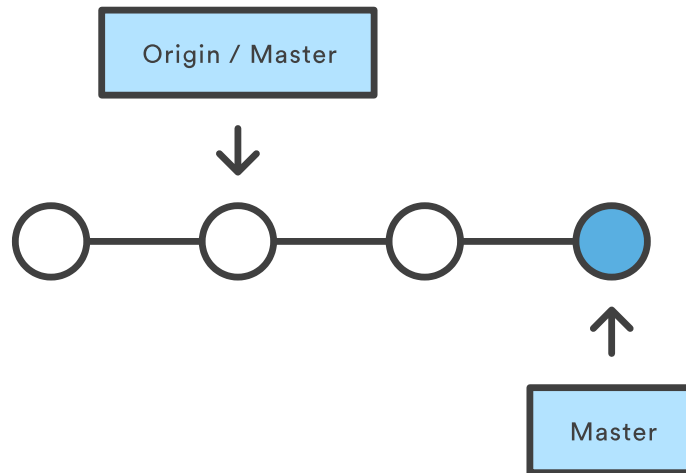
```
git push <remote> <branch>
```

Envía la rama especificada a una , junto con todos los commits y objetos internos necesarios. De este modo se crea una rama local en el repositorio de destino. Para evitar que se sobrescriban los commits, Git no te permitirá enviarlos cuando el resultado en el repositorio de destino sea una fusión sin avance rápido.

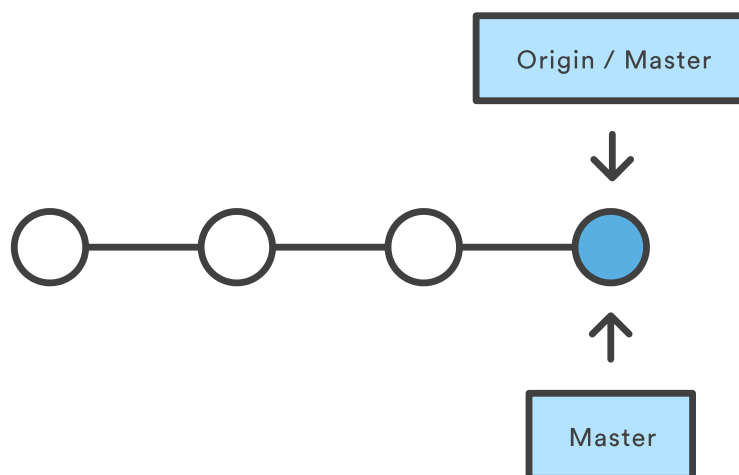
Análisis de git push

git push se suele usar para publicar y cargar cambios locales a un repositorio central. Después de modificar el repositorio local, se ejecuta un comando push para compartir las modificaciones con miembros remotos del equipo.

Before Pushing



After Pushing



El diagrama anterior muestra lo que pasa cuando el progreso de tu rama master local ha sobrepasado a la rama master del repositorio central y publicas los cambios mediante la ejecución de un comando `git push origin master`. Fíjate en que ejecutar `git push` es similar en su esencia a ejecutar `git merge master` desde el repositorio remoto.

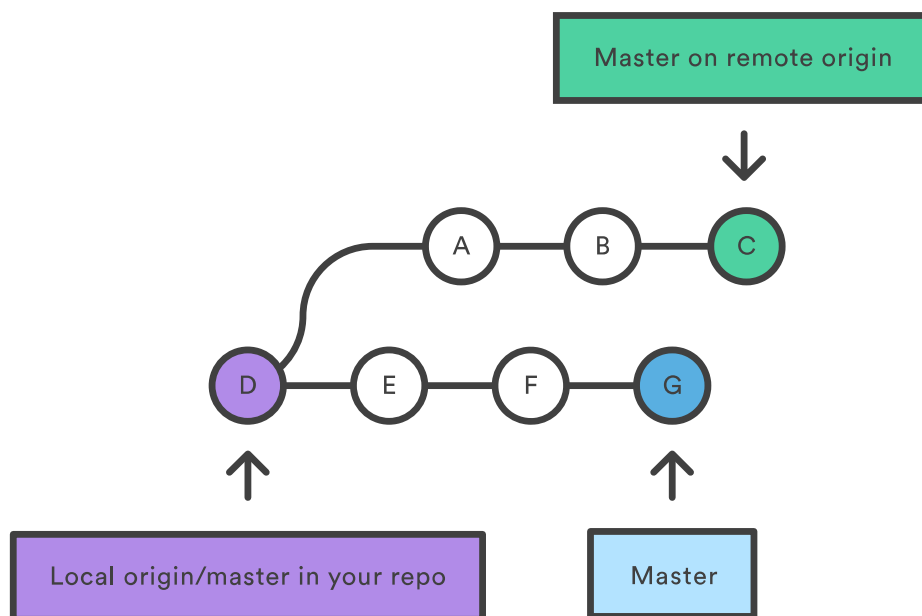
Uso de git pull

El comando `git pull` se emplea para extraer y descargar contenido desde un repositorio remoto y actualizar al instante el repositorio local para reflejar ese contenido. La fusión de cambios remotos de nivel superior en tu repositorio local es una tarea habitual de los flujos de trabajo de colaboración basados en Git. El comando `git pull` es, en realidad, una combinación de dos comandos, `git fetch` seguido de `git merge`.

Análisis de git pull

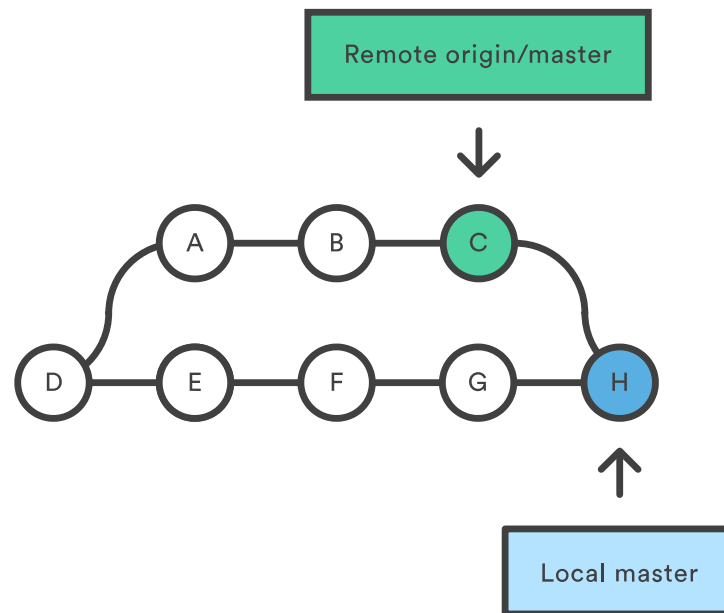
El comando `git pull` ejecuta en primer lugar `git fetch`, que descarga el contenido del repositorio remoto especificado. Después, se ejecuta `git merge` para fusionar las referencias y los encabezados del contenido remoto en una nueva confirmación de fusión local.

Para ilustrar mejor el proceso de incorporación de cambios y fusión, veamos el siguiente ejemplo. Supongamos que tenemos un repositorio con una rama maestra y un origen remoto.



En este escenario, `git pull` descargará todos los cambios desde el punto de separación de la rama local y la rama maestra. En este ejemplo, ese punto es E. `git pull` extraerá las confirmaciones remotas divergentes que son A, B y C.

A continuación, el proceso de incorporación de cambios creará una nueva confirmación de fusión local que incluya el contenido de las nuevas confirmaciones remotas divergentes.



En el diagrama anterior, podemos ver la nueva confirmación H, que es una confirmación de fusión nueva que incluye el contenido de las confirmaciones remotas A, B y C, y tiene un mensaje de registro combinado. Este es un ejemplo de una de las estrategias de fusión de `git pull`.

- Generacion de cambio en master.
- otro cambio en master 2.