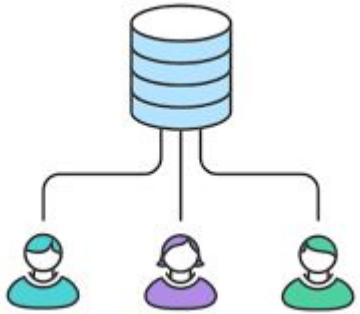
Introducción al Módulo 2

Los repositorios y archivos digitales de acceso abierto son espacios virtuales, con soporte de base de datos, en los que se puede depositar documentación científica de todo tipo y en todos los formatos posibles.

Repositorio Git

En Git un repositorio es un proyecto de software controlado por este sistema de control de versiones. Físicamente no es más que una carpeta del sistema de archivos con código fuente de una aplicación.

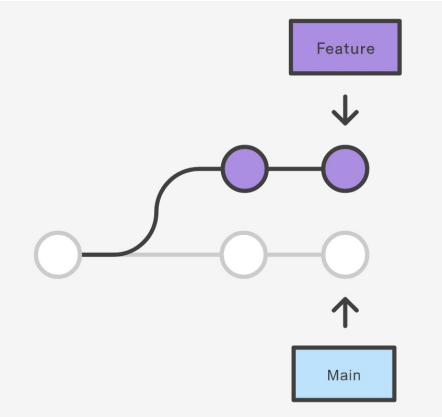


Por qué utilizar Git en tu organización

Pasar de un sistema de control de versiones centralizado a Git cambia la forma en que tu equipo de desarrollo crea software. Y si tu empresa depende de su software para aplicaciones críticas, la modificación del flujo de trabajo de desarrollo afecta a toda la empresa.

Flujo de trabajo

Una de las mayores ventajas de Git son sus capacidades de ramificación. A diferencia de los sistemas de control de versiones centralizados, las ramas de Git son baratas y fáciles de fusionar. Esto facilita el flujo de trabajo de ramas de función tan popular entre muchos usuarios de Git.

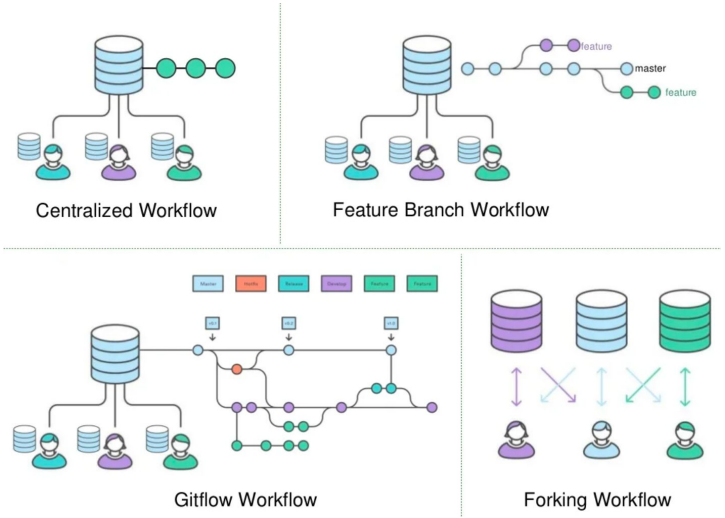


Las ramas de función proporcionan un entorno aislado para cada cambio en tu código base. Cuando un desarrollador quiere empezar a trabajar en algo, sin importar lo grande o pequeño sea, crea una nueva rama. Esto garantiza que la rama principal siempre contenga código de calidad para producción.

El uso de ramas de función no solo es más fiable que editar directamente el código de producción, sino que además proporciona ventajas organizativas. Te permite representar el trabajo de desarrollo con la misma granularidad que tu backlog ágil.

Control de versiones

El control de versiones, también conocido como "control de código fuente", es la práctica de rastrear y gestionar los cambios en el código de software. Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo.

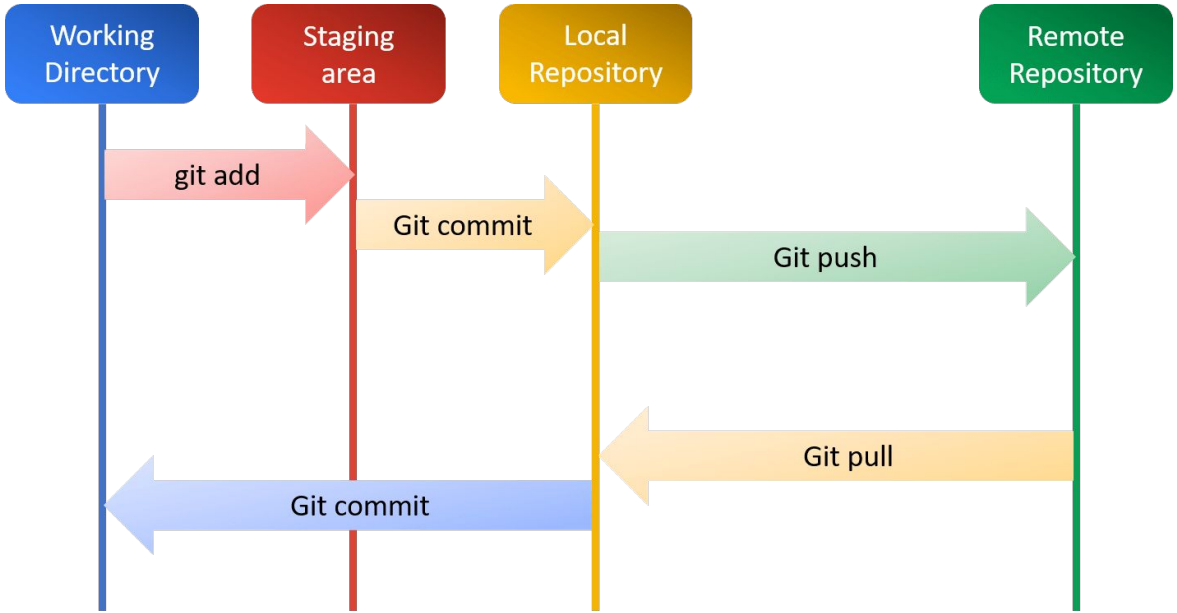


Fuente: Click aquí

Comandos GIT

Comandos git más frecuentes:

- git clone
- git checkout
- git add
- git status
- git commit
- git push
- git pull
- git merge



Fuente: Click aquí

Git clone

Descarga un repositorio a nuestra computadora

Syntax of git clone

```
git clone < Repository URL>
```

Git add

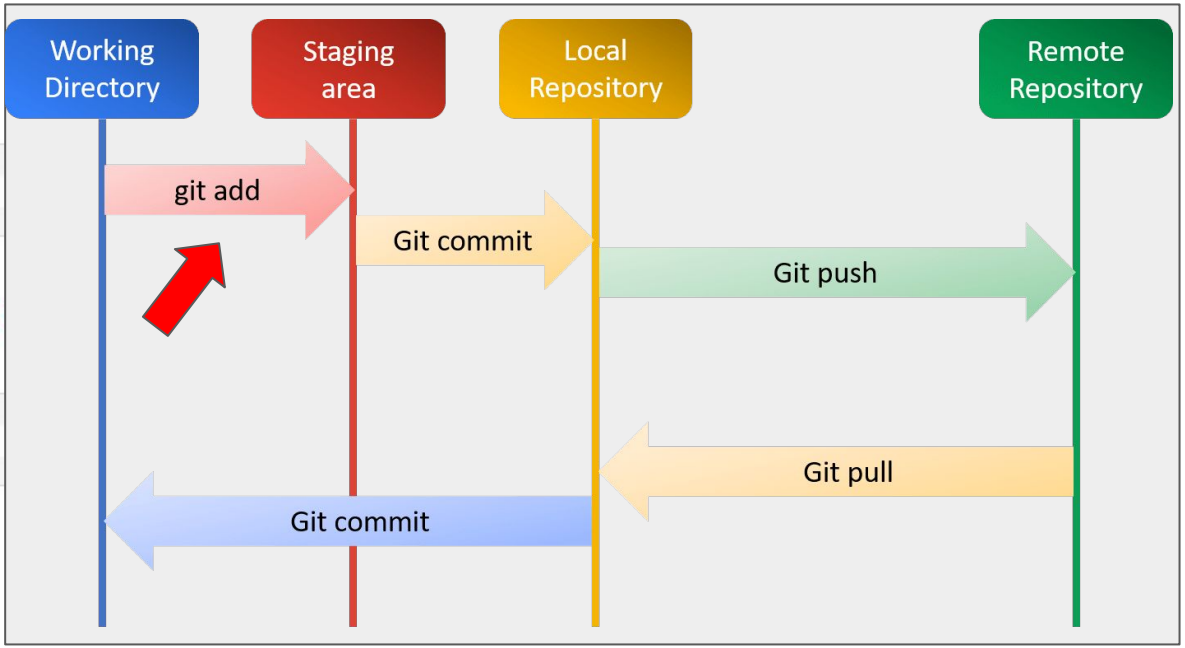
Agrega archivos modificados a área de trabajo local “staging”

Syntax to add all modified file

```
git add .
```

Syntax to add single modified

```
git add <File Name with full path>
```

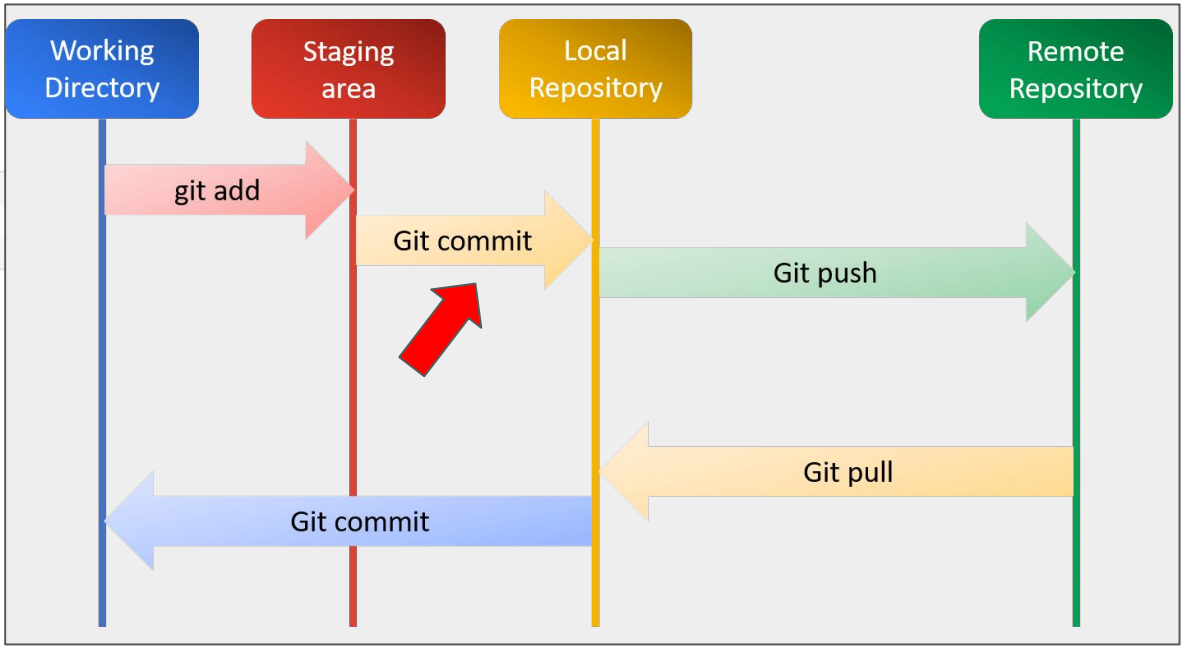


Git commit

Agrega los cambios del área “staging” al “repositorio local”

Syntax of git commit

```
git commit -m "add commit message here"
```

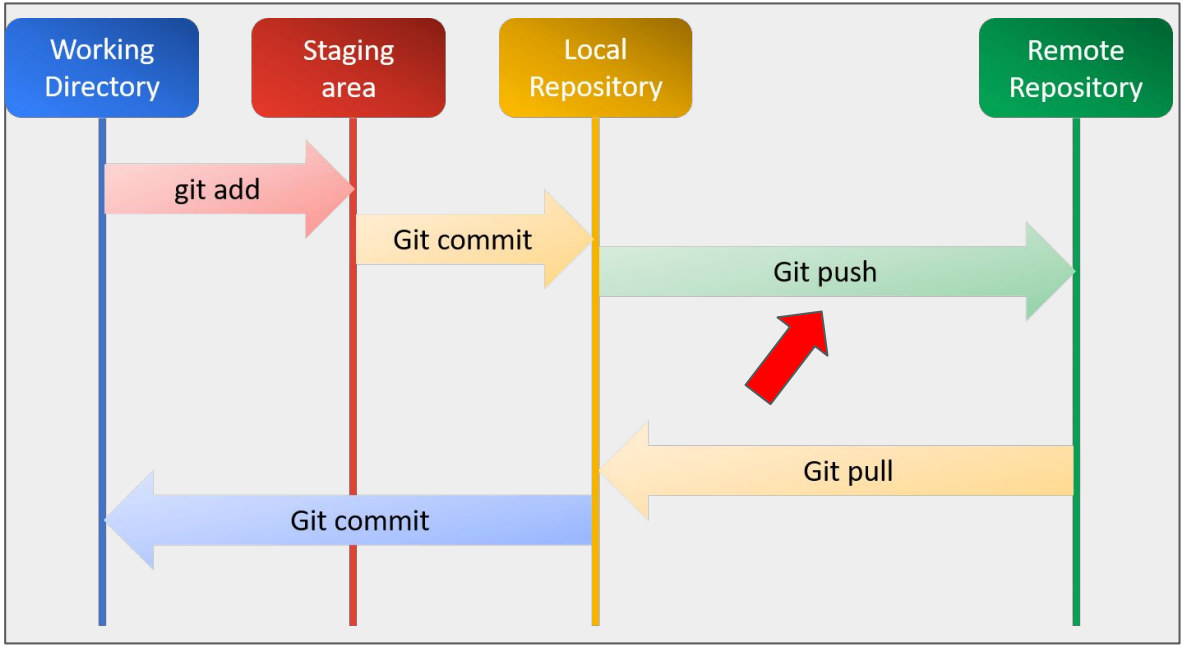


Git push

Sube los cambios desde “repositorio local” al “repositorio remoto”

Syntax of git push

```
git push
```

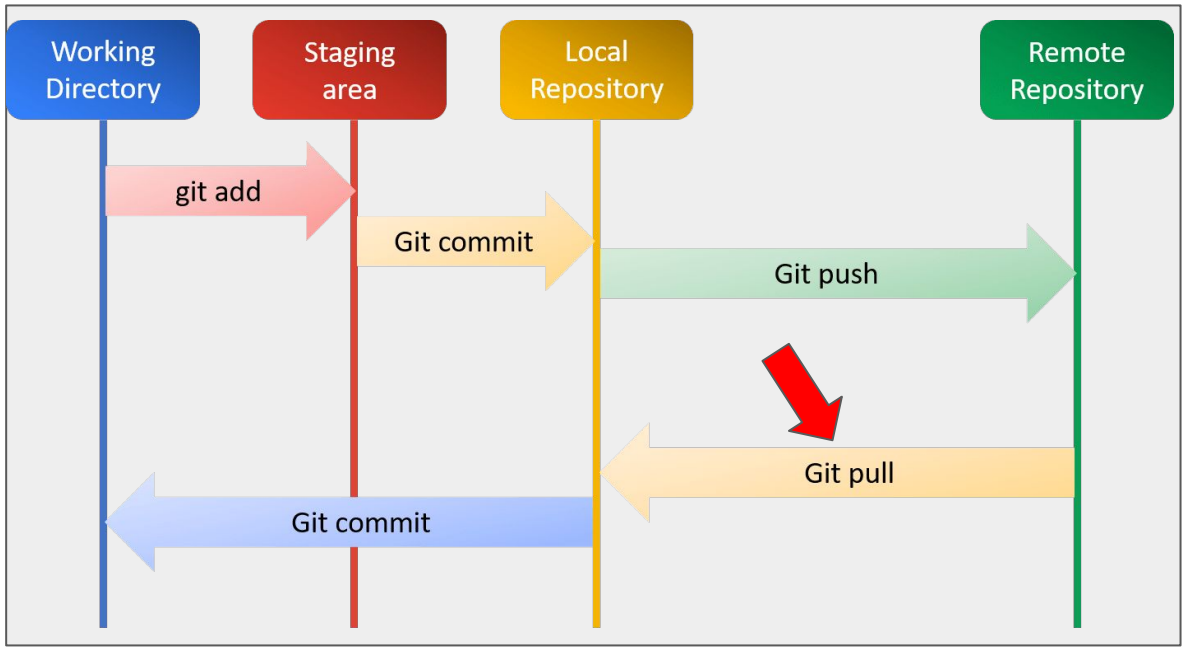


Git pull

Descarga cambios desde “repositorio remoto” al “repositorio local”.
Usado para recuperar cambios realizados por los demás desarrolladores

Syntax of git pull

```
git pull
```



Git status

Nos muestra los archivos modificados en “staging” y “no staging” área después del ultimo commit

Syntax of git status

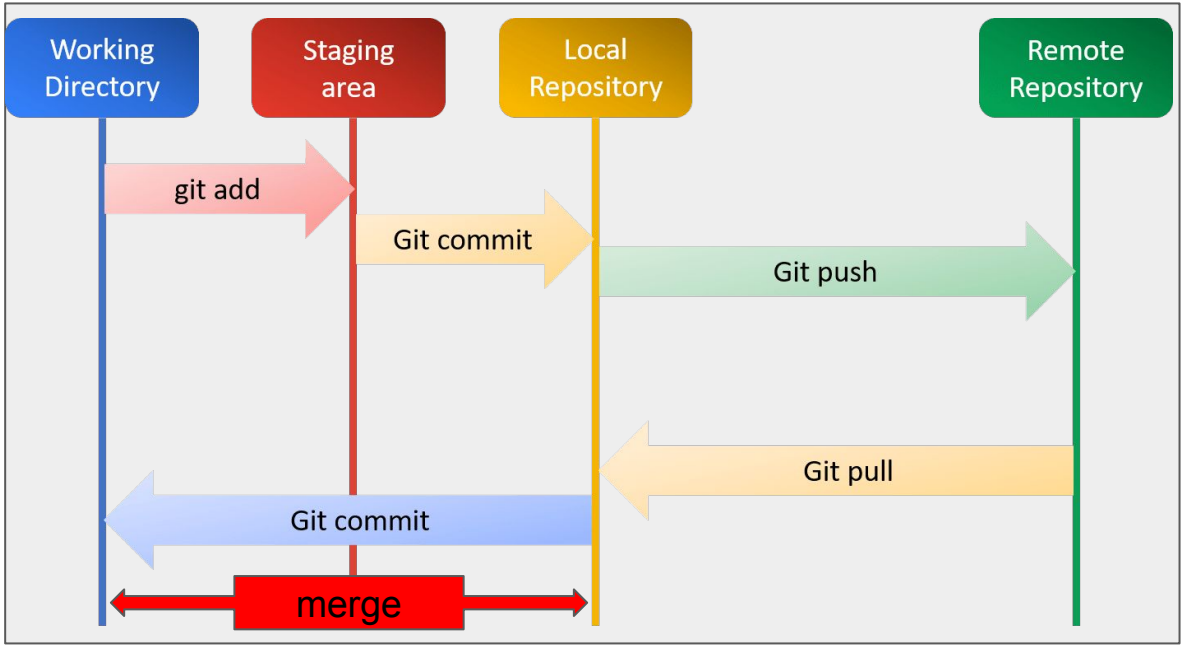
```
git status
```

Git merge

Une cambios entre “repositorio local” y “working directory”

Syntax of git merge

```
git merge <commit id>
```



Git checkout

Nos permite cambiarnos de rama del código

Syntax of git checkout

```
git checkout <Branch Name>
```