



Git

FRONTEND INICIAL

FRONTEND INICIAL

FRONTEND INICIAL





OBJETIVO DEL TEMA

- Que es Git
- Instalar Git
- Configurar credenciales de git en tu PC
- Como usar Git (Comandos)
- Estados dentro de git
- Repositorio Local y Remoto (Estructura)

**¿Qué es
Git?**

**Sistema Control
de Versiones**

**Regresar en el
Tiempo**

**Tener copias
paralelas (branch)**

Git
NO ES
Github

Comandos



Que es Git?

Git es un sistema de control de versiones gratuito y de código abierto, diseñado para manejar desde pequeños a grandes proyectos de manera rápida y eficaz.

Se entiende como control de versiones a todas las herramientas que nos permiten hacer modificaciones en nuestro proyecto. Este sistema registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo.

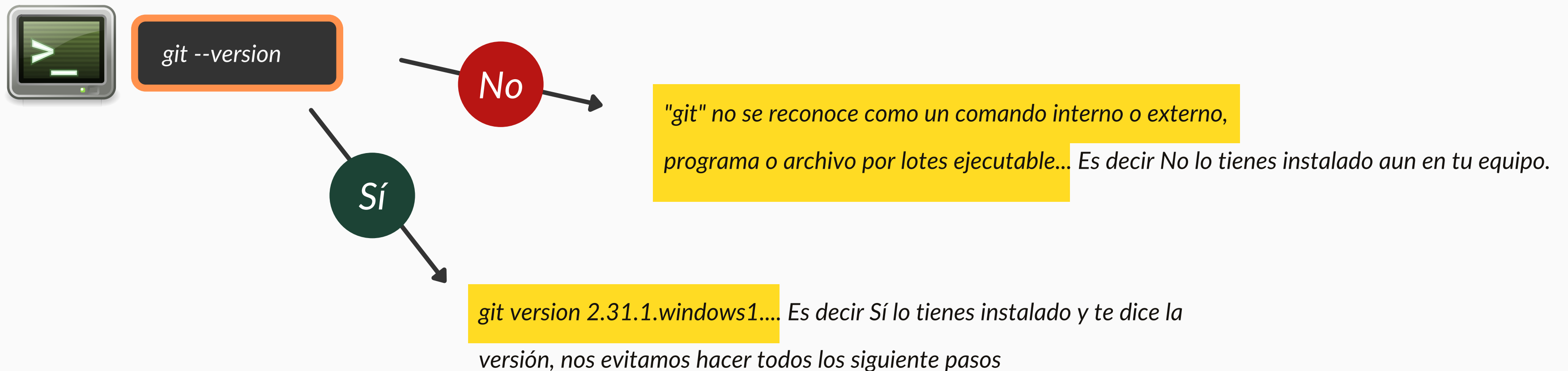
[Pagina Oficial](#)



Previo a instalar Git?

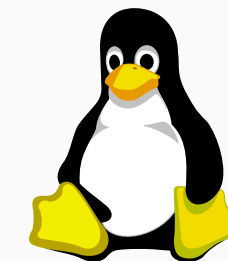
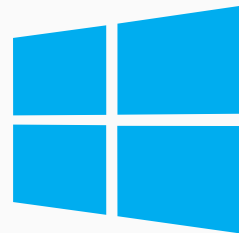
En primera instancia podemos comprobar si ya tenemos o no instalado "git" en nuestro ordenador.

Abriremos la terminal (consola) que tenemos por defecto en cualquier sistema operativo donde estemos trabajando:



Instalar Git?

*Git funciona en cualquier sistema operativo
haz clic sobre el sistema operativo para ver la guía de instalación*



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



Windows: Cuando instalamos git viene acompañado de un terminal adicional llama "Git Bash"

Configurar credenciales en git

Debemos abrir la terminal



Vamos a configurar tanto "nombre y apellido" como "tu-correo@gmail.com"

Como lo hacemos?

```
git config --global user.name "Nombre y Apellido"
```

```
git config --global user.email "tu-correo@gmail.com"
```

Como lo verificamos?

```
git config user.name
```

```
git config user.email
```



Esto se hace una única vez por equipo y también lo podemos modificar en cualquier momento

Inicializar git

Por cada proyecto que estemos desarrollando y queramos implementar o inicializar "git", debemos hacer lo siguiente

IMPORTANTE

Debemos posicionarnos o asegurarnos que estemos dentro la carpeta que queremos colocar "git" desde la terminal

Ejecutar `git init` desde la terminal

A screenshot of the Visual Studio Code interface. The terminal window at the bottom shows the command prompt for Windows. The user has entered 'git init' and the output is 'Initialized empty Git repository in C:/Users/leona/Desktop/mi-proyecto/.git/'. The status bar at the bottom indicates the current branch is 'master'.

```
Microsoft Windows [Versión 10.0.19043.1110]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\leona\Desktop\mi-proyecto>git init
Initialized empty Git repository in C:/Users/leona/Desktop/mi-proyecto/.git/

C:\Users\leona\Desktop\mi-proyecto>
```



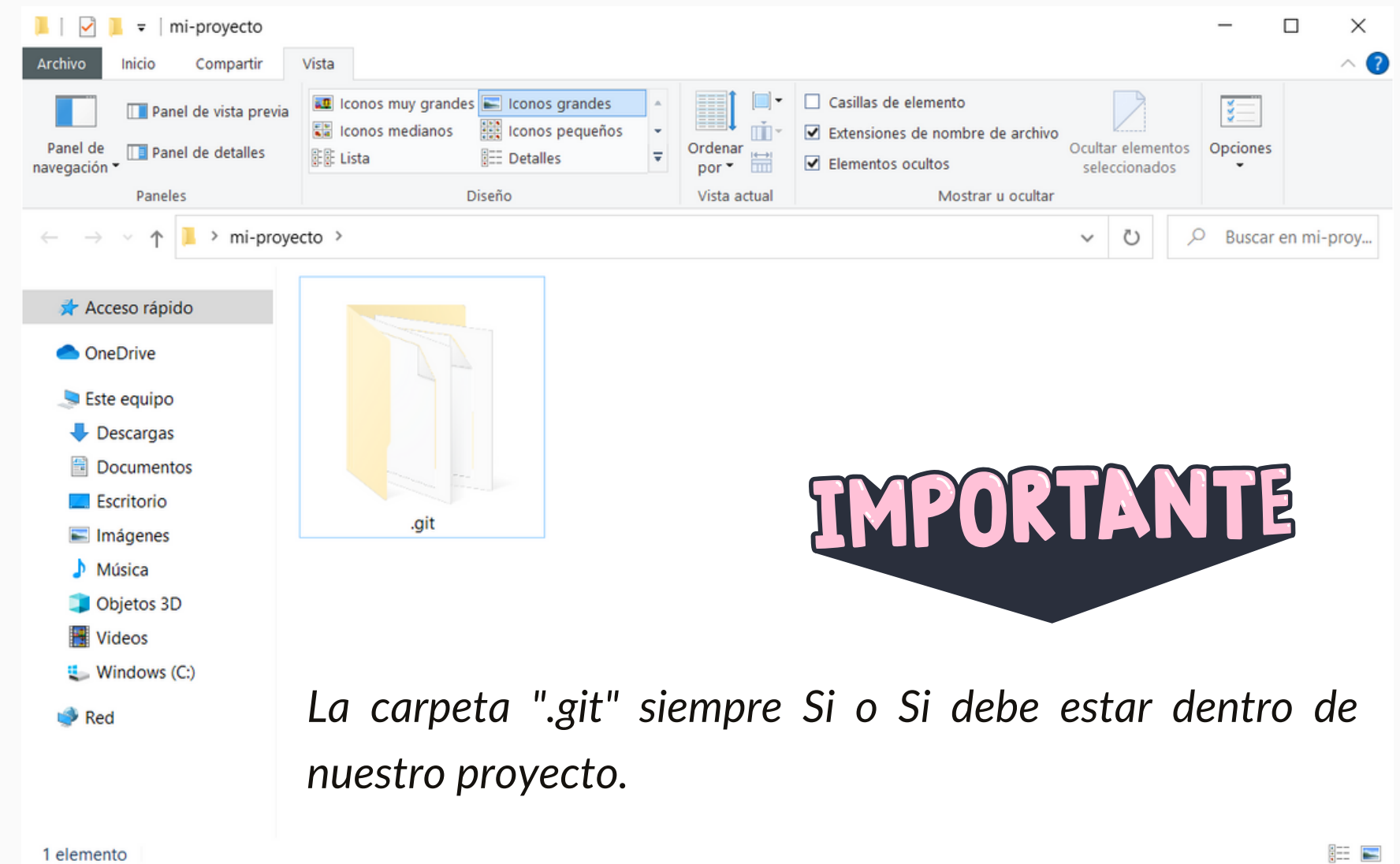
Recuerda reiniciar el editor de código o terminales al terminar la instalación para que te reconozca "git"

Verificar proyecto con git

Si nos vamos al explorador de archivos y vemos dentro de nuestra proyecto esta una carpeta oculta llamada ".git", va todo OK.

Aclaraciones o Advertencias:

- *No debe existir otra carpeta llamada "git" o ".git" dentro del proyecto para evitar inconvenientes.*
- *Asegurarnos que estamos dentro de la carpeta que estaremos trabajando o desarrollando.*
- *Tener claro como moverte dentro tus carpetas y archivos y saber donde estas posicionado te ayudara.*



IMPORTANTE

La carpeta ".git" siempre Si o Si debe estar dentro de nuestro proyecto.



Para poder ver elementos ocultos desde el explorador, ir al panel superior "vista" y tildar "Elementos ocultos"

Comandos básicos en git

Consultar el estado de tus archivos y código dentro del proyecto y rama.

```
git status
```

Preparar y pasar todos los cambios al segundo estado dentro del repositorio local llamado "staging area".

```
git add .
```

Generar un nuevo commit o versión dentro del repositorio local.

```
git commit -m "Tu mensaje descriptivo"
```

Listar los commits dentro del repositorio local.

```
git log
```

Crear una nueva rama dentro del repositorio local.

```
git branch nueva-rama
```

Moverme a una rama distinta.

```
git switch nueva-rama
```

Comandos básicos en git (2)

Moverme a un commit, versión anterior para consultar sin borrar commits.

```
git checkout head~Num
```

Para regresarme al commit mas reciente del repositorio local.

```
git checkout master
```

Moverme a un commit, versión anterior para consultar borrando commits sobrantes.

```
git reset --hard head~Num
```

Eliminar una rama.

```
git branch -d borrar-rama
```

Modificar nombre de una rama.

```
git branch -M nombre-disinto-rama
```

Comprobar si tenemos enlazado el repositorio local con repositorios remoto.

```
git remote -v
```

Comandos básicos en git (3)

Vincular repositorio local con repositorio remoto.

```
git remote add origin https://github.com/usuario/mi-proyecto.git
```

Subir, empujar los cambios al repositorio remoto, te puede llegar a pedir autorizar credenciales de tu cuenta de github.

```
git push -u origin master
```

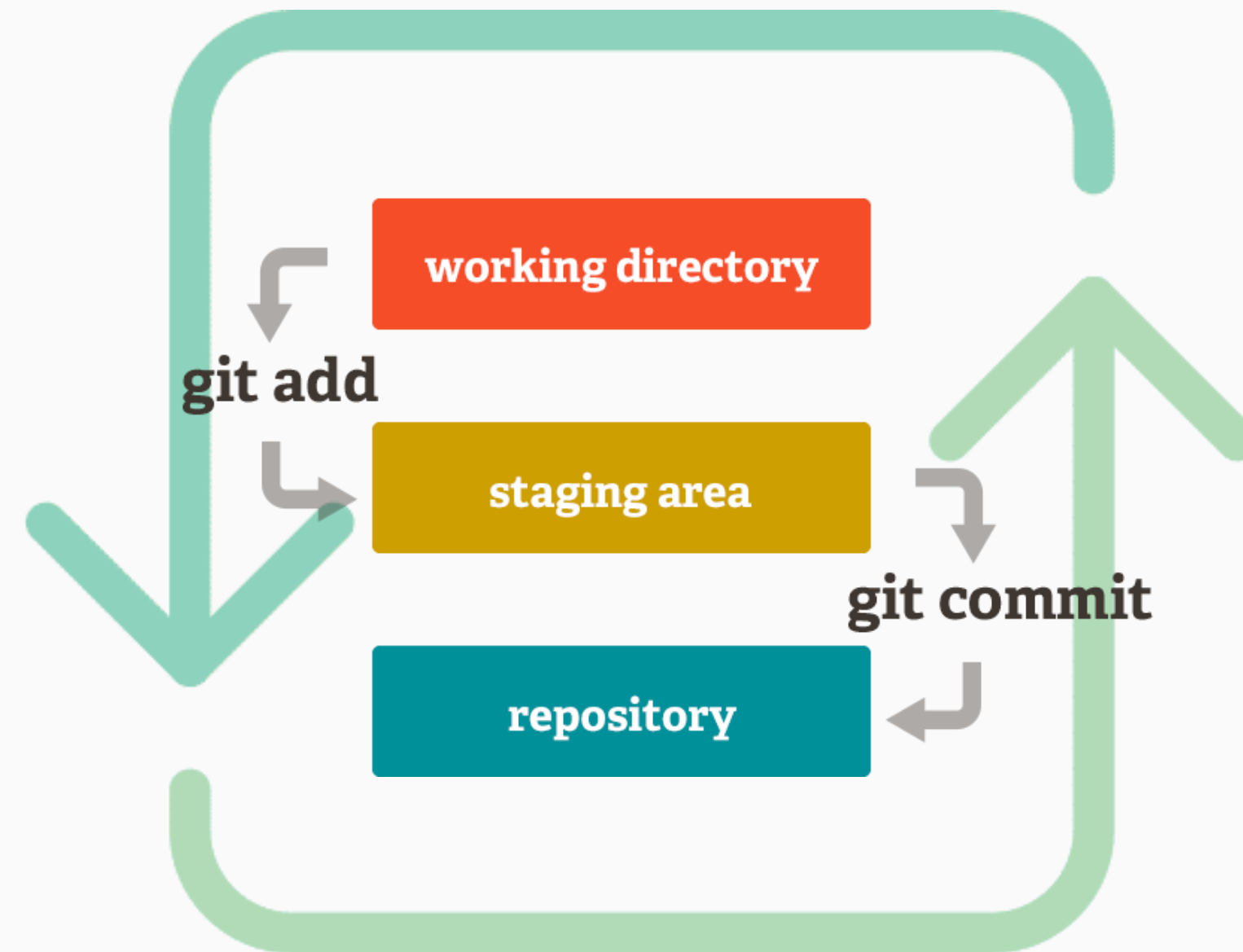
halar, traer los cambios al repositorio local, en algunas ocasiones te puede pedir resolver conflictos.

```
git pull origin master
```

Fusionar o combinar ramas, por lo general debes estar en la rama principal donde quieres unir los cambios o commits de la rama secundaria

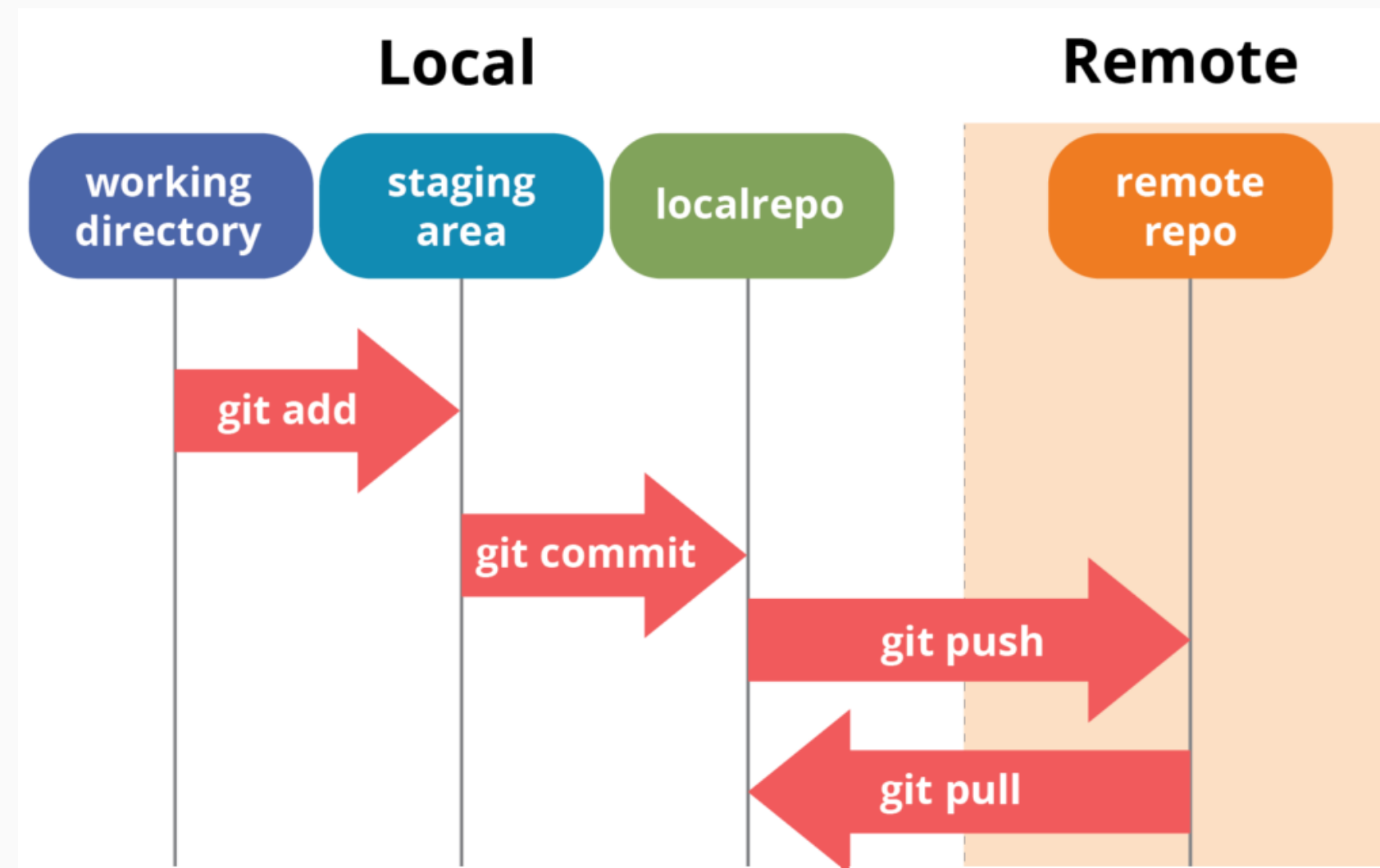
```
git merge rama-secundaria
```


Estados de Git



Recordar como mínimo tener una primera version o commit y así tener de donde partir en el historial de versiones de tu proyecto

Repositorio Local y Remoto



Flujo de trabajo normal cuando estas desarrollando tanto de manera local como de manera colaborativa