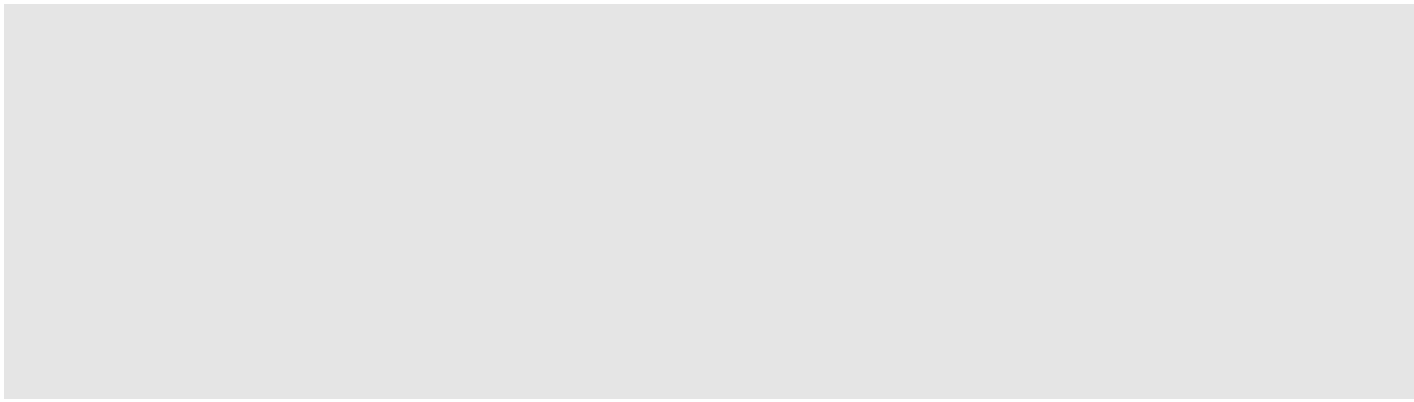


Introducción a Git



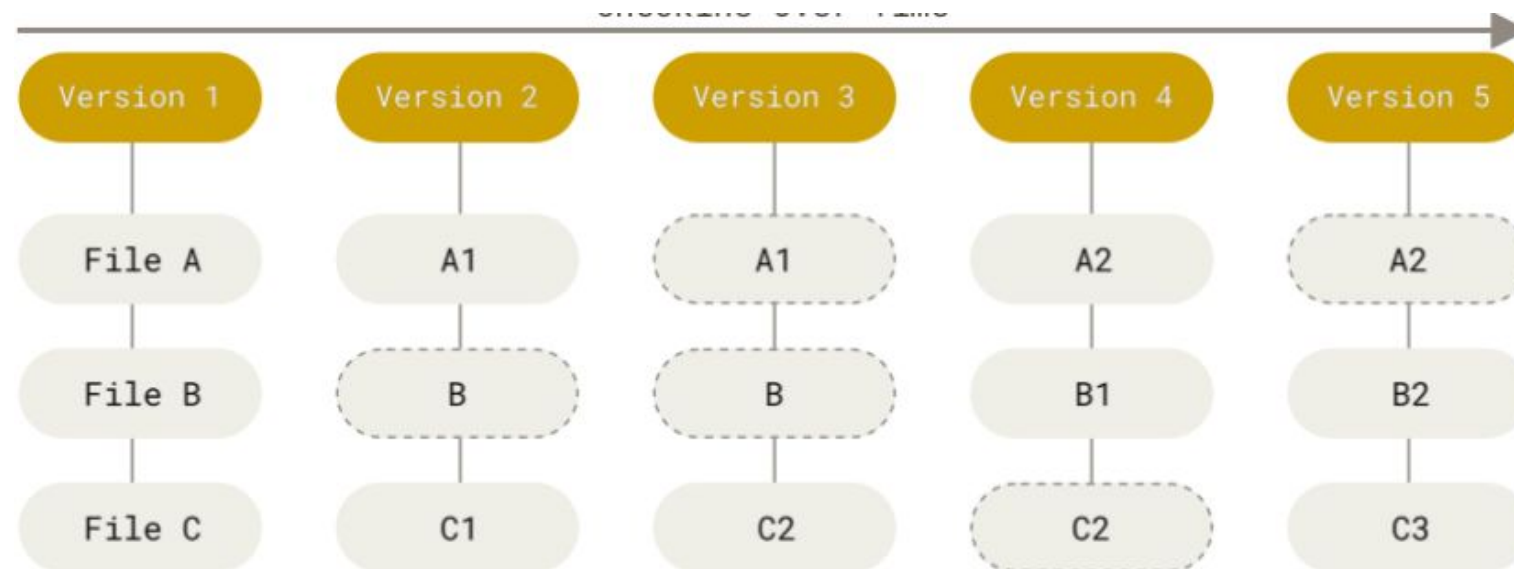
Controlador de versiones

Herramienta de Software que ayuda a gestionar los cambios en archivos de código. Las principales ventajas que deberías esperar del control de versiones son las siguientes.

- Provee un historial completo de cambios a largo plazo de todos los archivos
- Permite creación de ramas y fusiones
- Debe ser capaz de trazar cada cambio anotando un mensaje que describa el propósito y el objetivo del cambio

GIT

Git es un sistema de control de versiones distribuido diseñado para manejar cualquier tipo de proyecto, ya sea grande o pequeño, con velocidad y eficiencia. Git realiza esto haciendo “snapshots” del proyecto.



Configuración

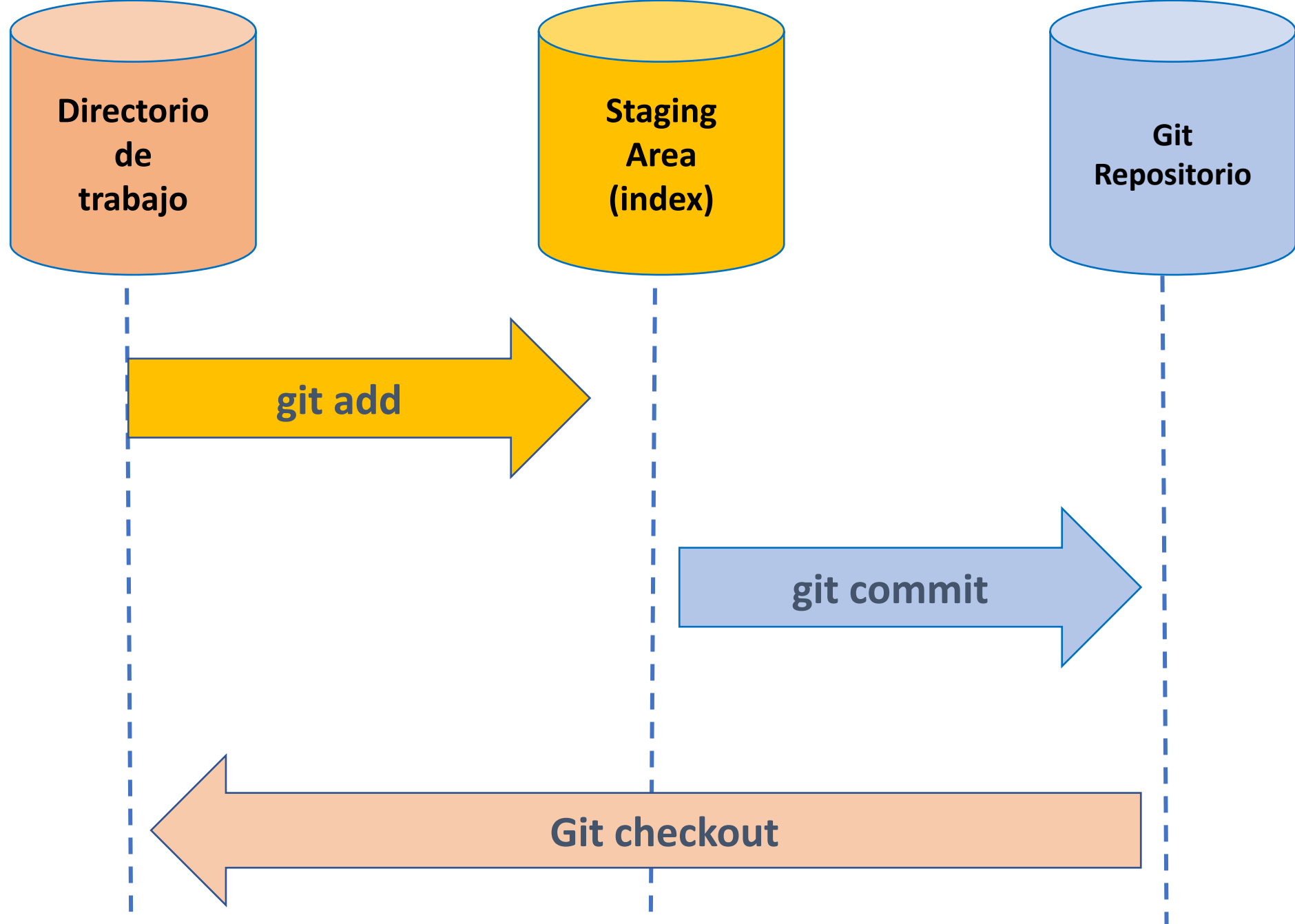
Definiendo nombre de usuario y dirección de correo electrónico

- ***git config --global user.name "username"***
- ***git config --global user.email "email"***

Verificando usuario

- ***git config user.name***
- ***git config user.email***

Comandos comunes

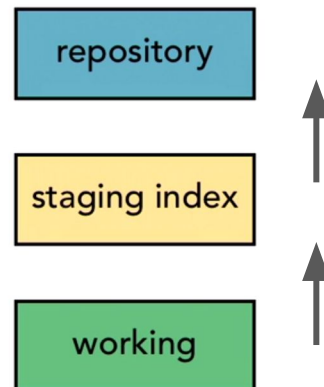


Tres estados de tus archivos

- Modificado (modified)
- Preparado (staged)
- Confirmado (committed)

Tres áreas de trabajo principales

- Repositorio
- Área de preparación
- Área de trabajo



Comandos Git - Creación del repositorio

git init <directory>

El comando crea un subdirectorio de .git que contiene todos los metadatos del nuevo repositorio. NO BORRAR

git clone <repo> <directory>

Crea un repositorio a partir de otro

Comandos Git - Configuración

Para guardar los cambios en git, es necesario identificar a los autores.

```
git config --global user.name "username"
```

```
git config --global user.email "email"
```


Comandos Git - Cambios

git status

Muestra el estado del directorio de trabajo y del staging área

git add <archivo /directorio>

Prepara todos los cambios antes del siguiente commit

git commit

Captura una instantánea de los cambios preparados en ese momento

Historial

git log

Muestra el historial de los commits realizados

Opciones útiles

-(n)	Muestra solamente las últimas n confirmaciones
--since, --after	Muestra los commits realizados después de la fecha especificada.
--until, --before	Muestra los commits antes de la fecha especificada.
--author	Muestra sólo aquellos commits realizados por un autor específico.
--grep	Muestra sólo aquellas confirmaciones que dentro del mensaje de confirmación contengan la palabra de búsqueda.

Visualizar cambios

git diff

Muestra todos los cambios sin confirmar desde la última confirmación

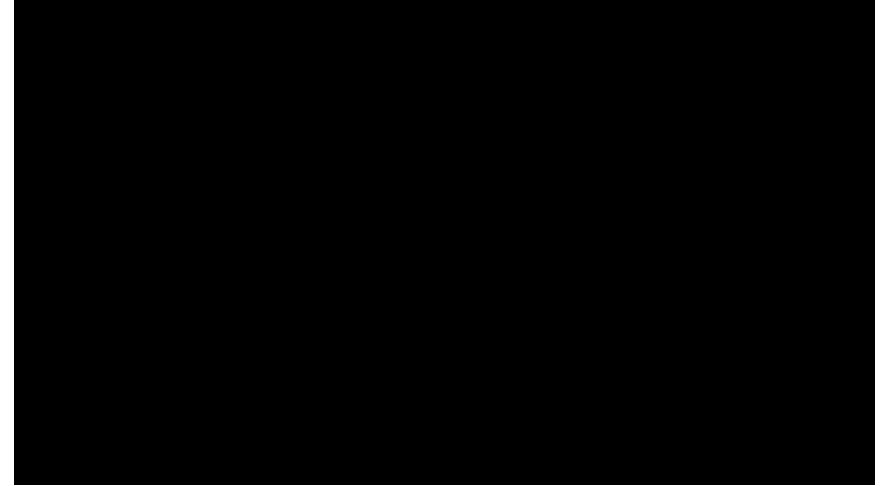
git diff HEAD <archivo>

Muestra todos los cambios de un archivo específico



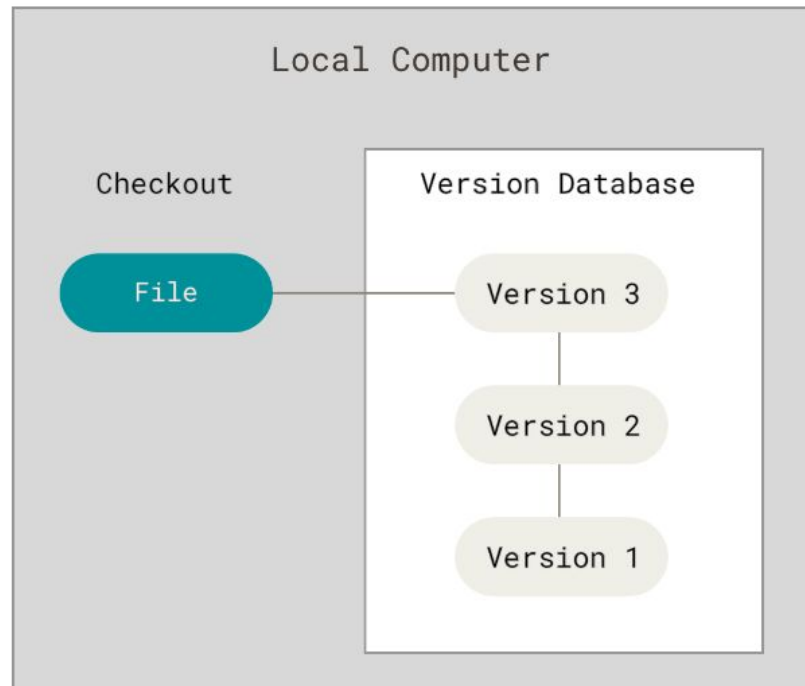
Sistema de control de
versiones

Vs

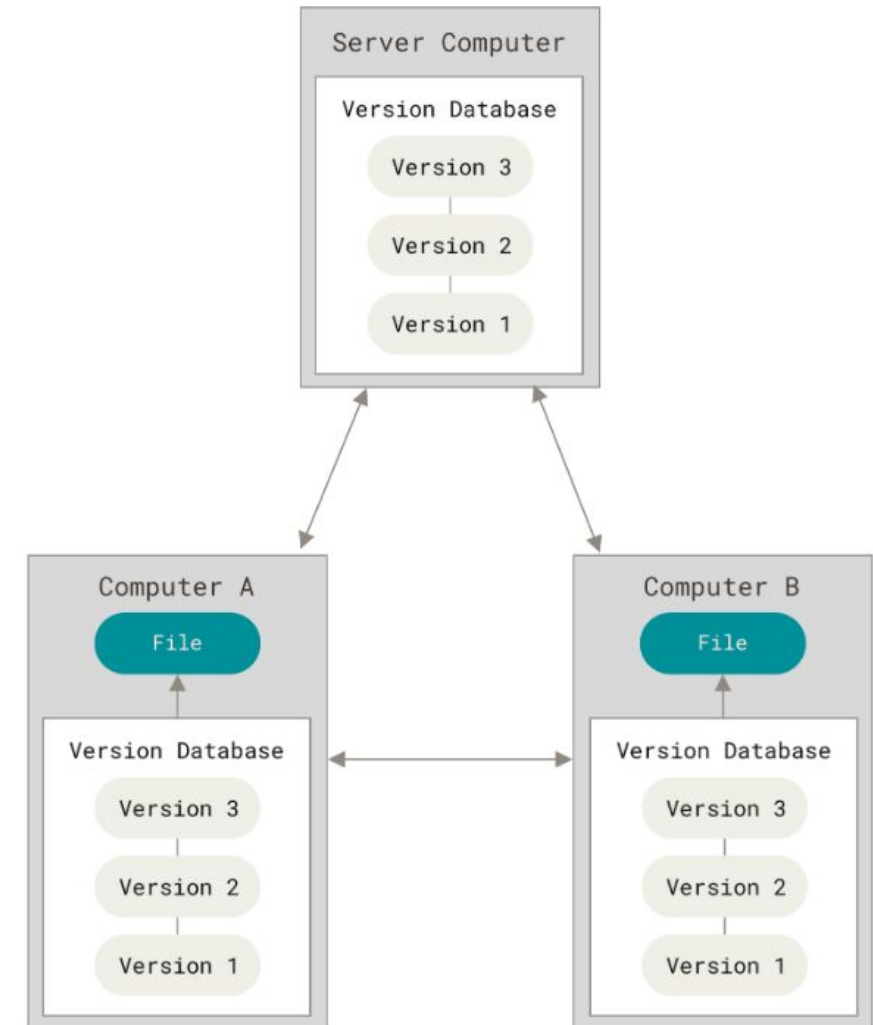


Servicio de alojamiento de
repositorios

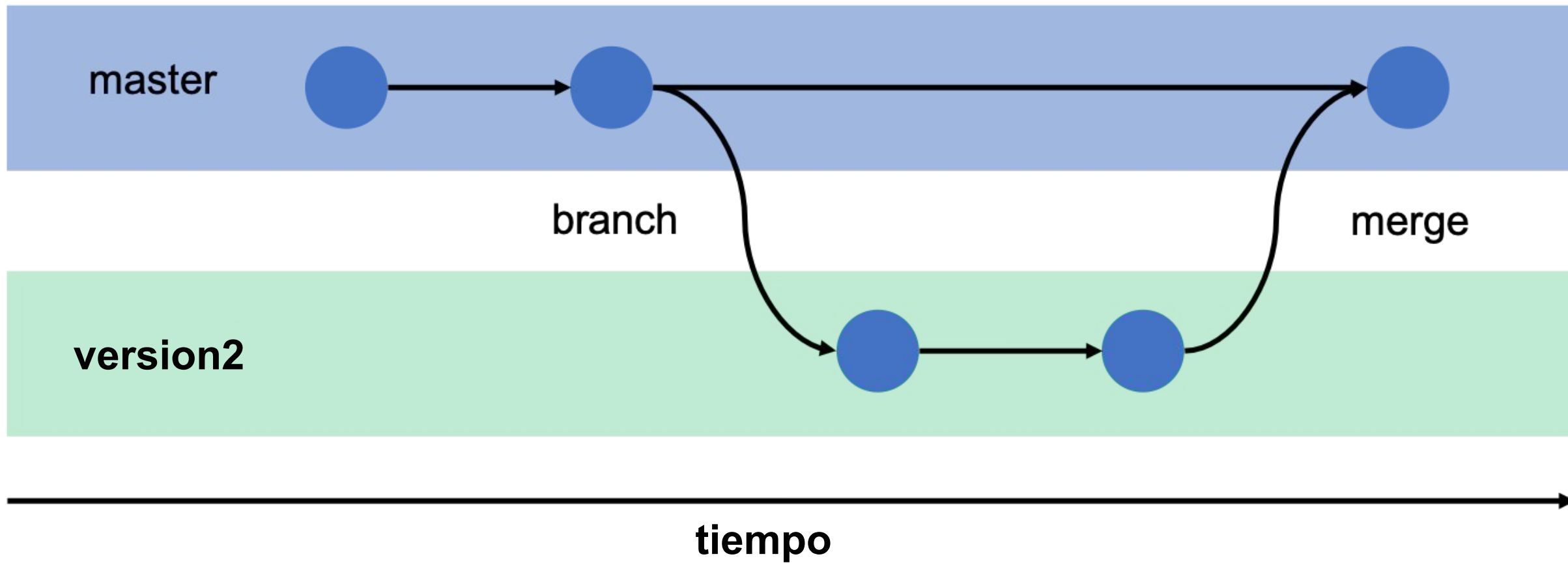
Control de versiones local



Control de versiones distribuido



Ramas



Branches

git branch <nombre>

Crea una rama

git checkout <rama>

Se usa para cambiar entre ramas

git merge <rama>

Integra los cambios de una rama

git branch -d <rama>

Elimina una rama