Taller Git y GitHub

Que es Git y como usarlo en GitHub para realizar proyectos

Importante antes de partir el Taller

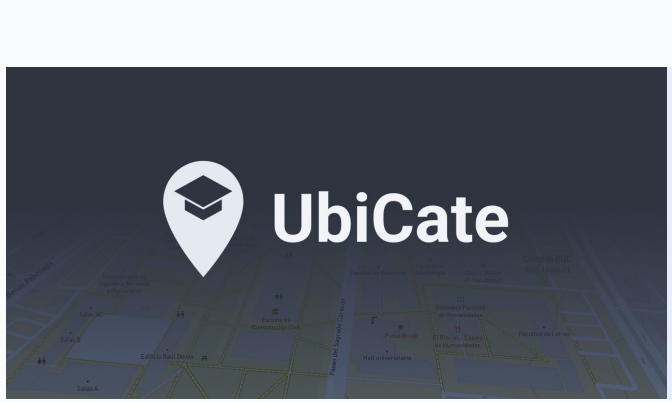
- La actividad requiere que hagan un setup previo.
- Todo el material del taller se encuentra en GitHub.
  - github.com/open-source-uc/taller-git-y-github-2023-2
- Vamos a asumir que estás algo familiarizado con un terminal.

Sobre el Taller

- Introducción a Git y GitHub + Actividad con lo aprendido
- Fuertemente inspirado en **CS50W** y Pro Git
- Hay ayudantes que podrán responder dudas durante el taller, ¡aprovechen de preguntar!
- La idea es llegar rápido a la actividad 🏃

Sobre Open Source UC

- Comunidad apasionada por crear e impulsar diversos proyectos de código abierto.
- Puedes unirte en proyectos guiados o a impulsar proyectos.
- Más info osuc.dev y en @opensource_euc en Instagram.



GitHub Campus Expert

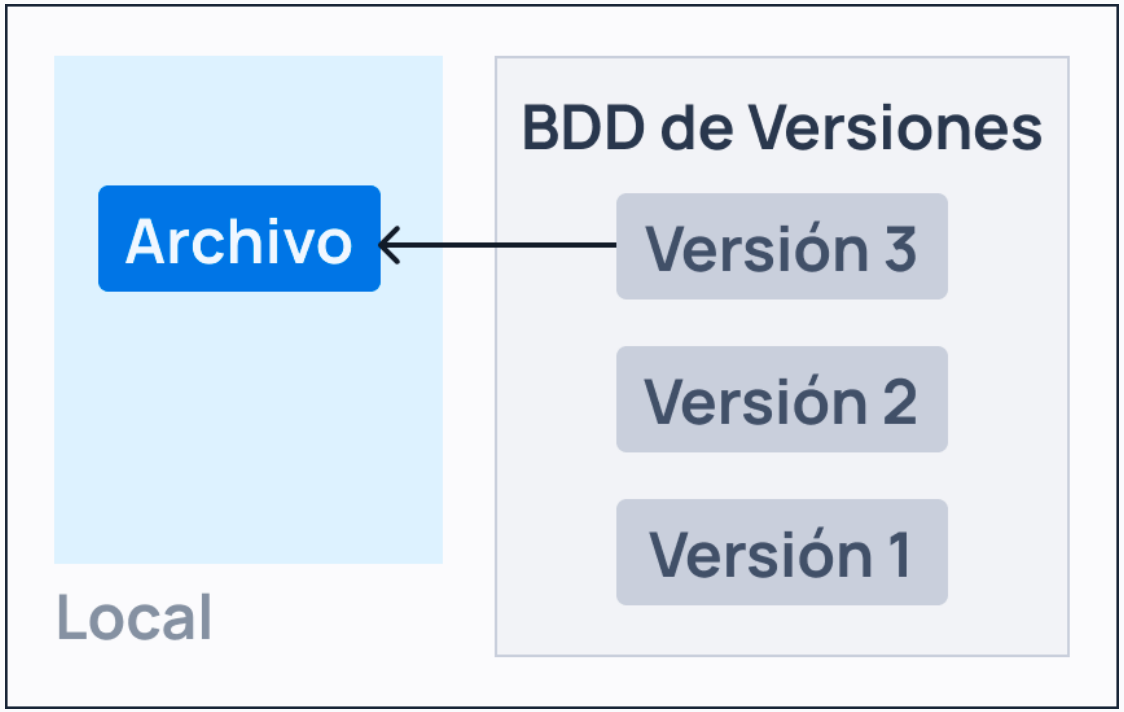
Campus Experts es un programa de GitHub de líderes estudiantiles que ayudan a comunidades de tecnología a ser inclusivas, diversas y con perspectiva a futuro con el power de GitHub:D

@IgnacioPalma

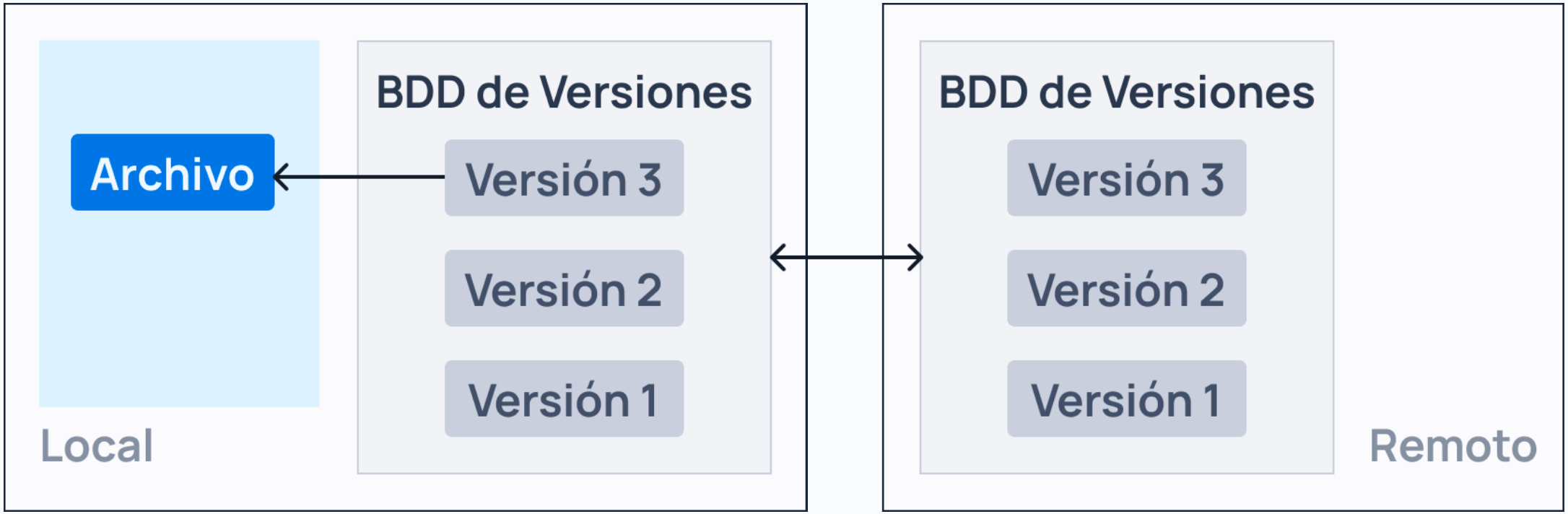
Founder of The Artificial Intelligence Club & GitHub Campus Expert

Qué es Git

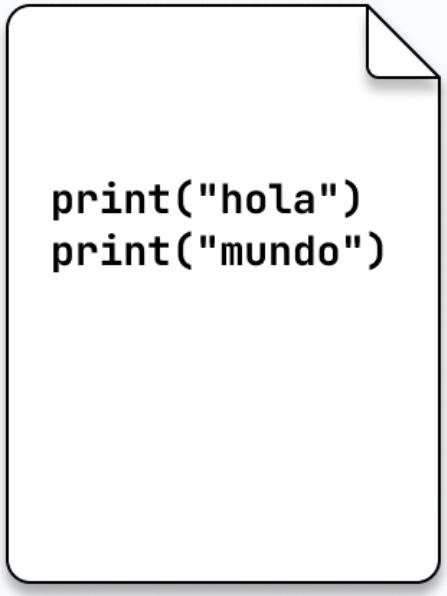
Qué es Git



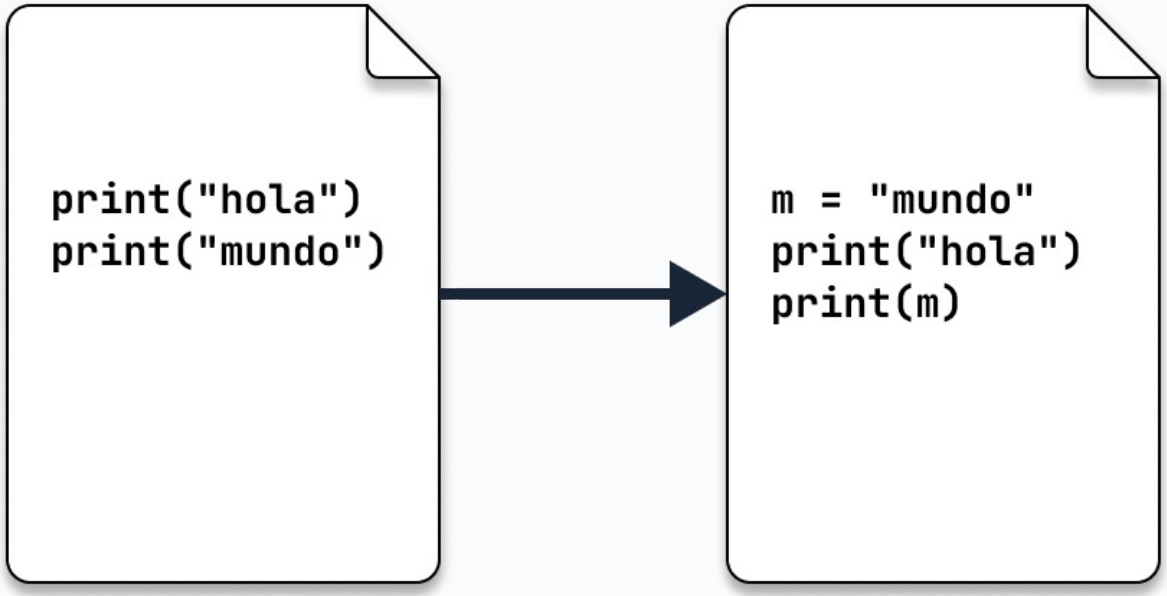
Qué es Git



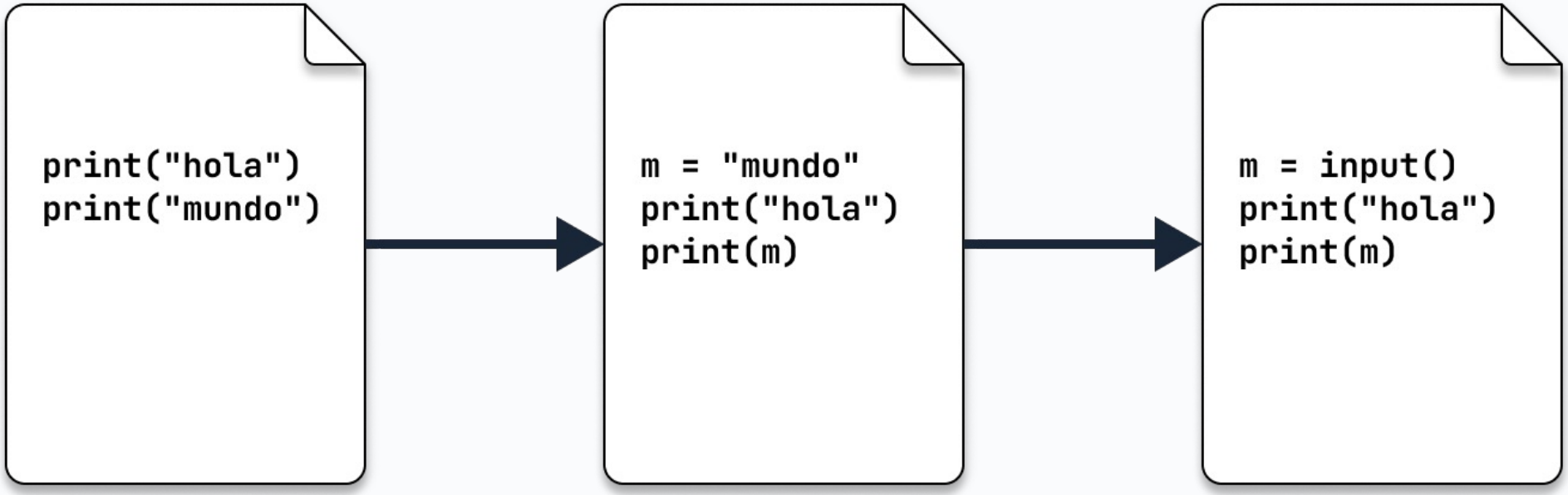
Mantiene versiones de archivos



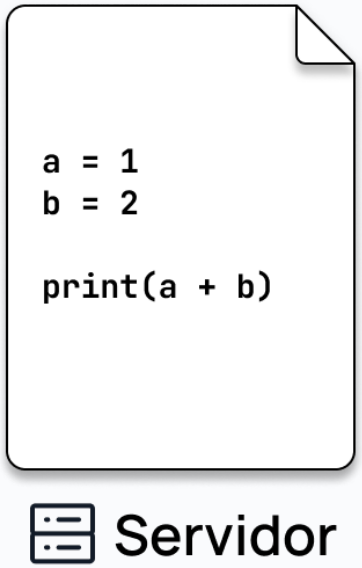
Mantiene versiones de archivos



Mantiene versiones de archivos



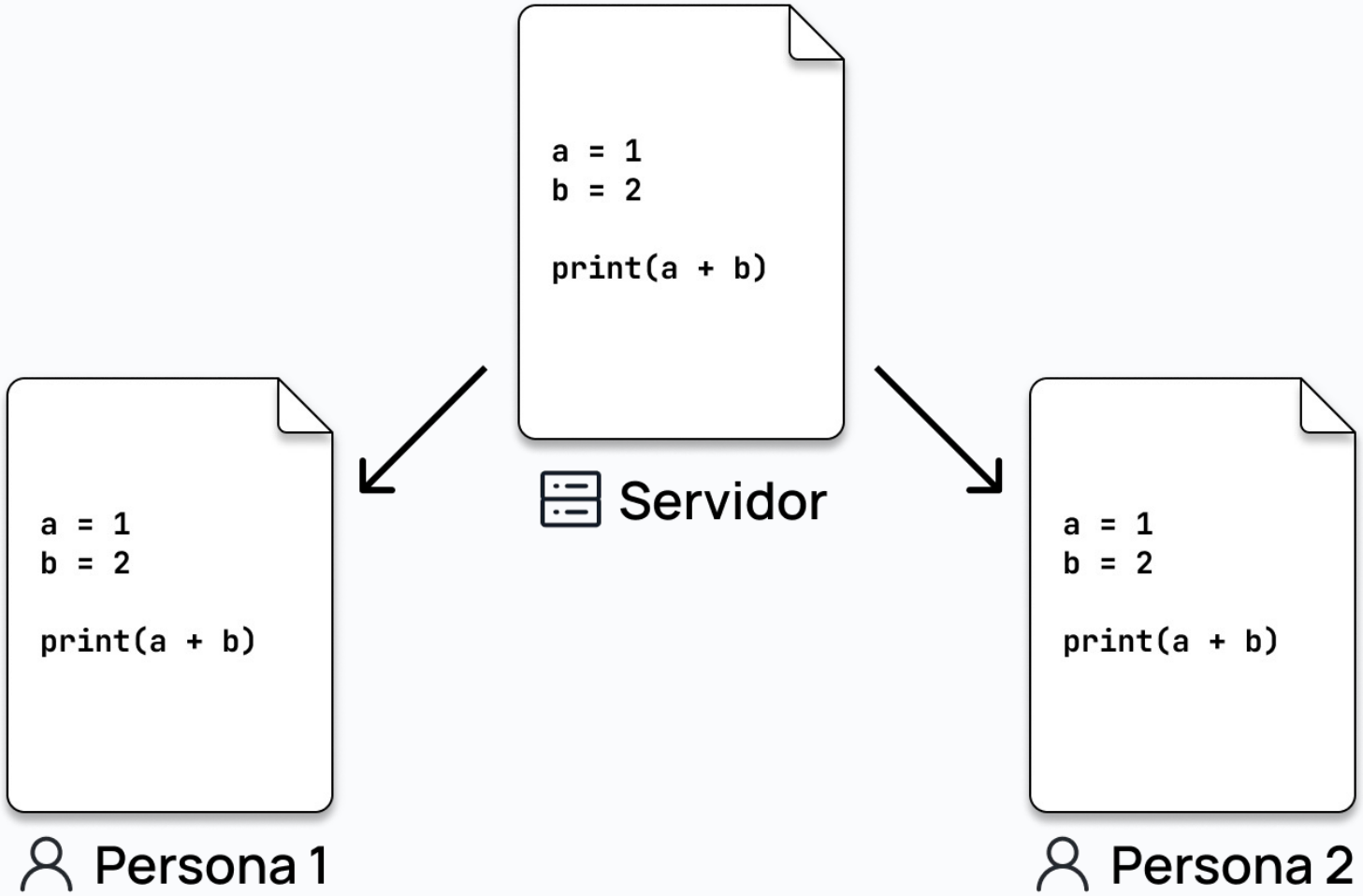
Sincronizar código entre personas



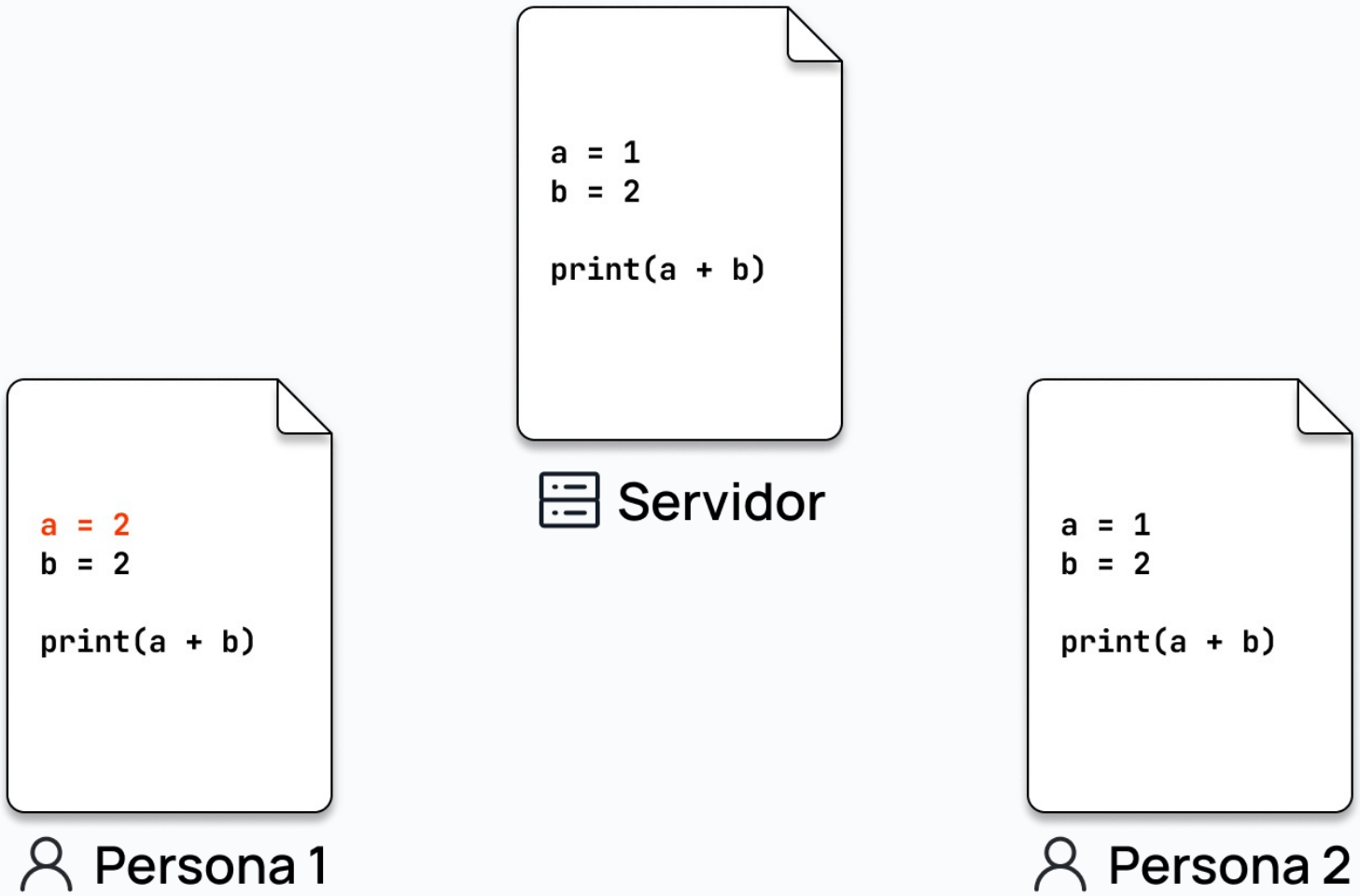
Persona 1

Persona 2

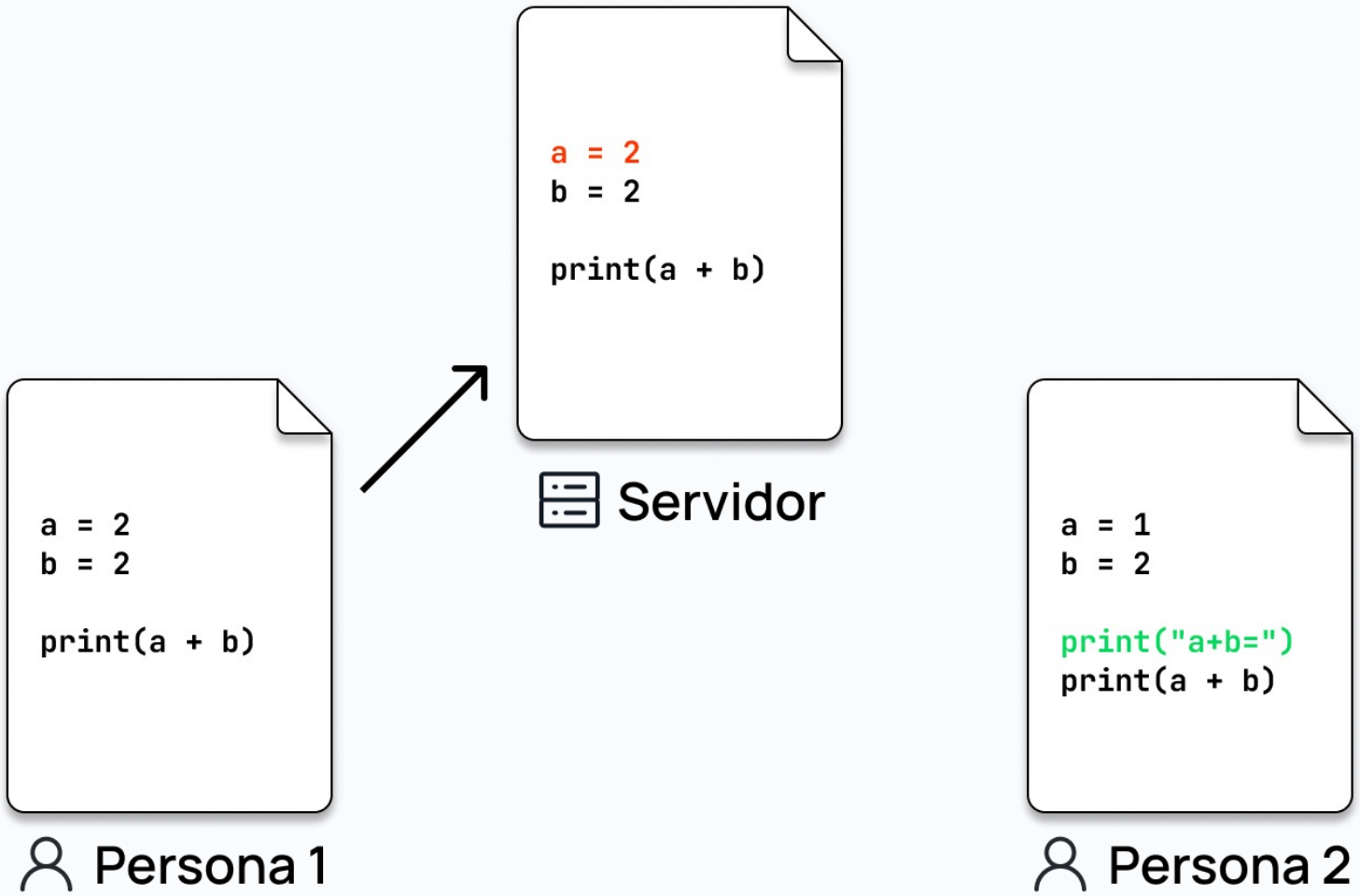
Sincronizar código entre personas



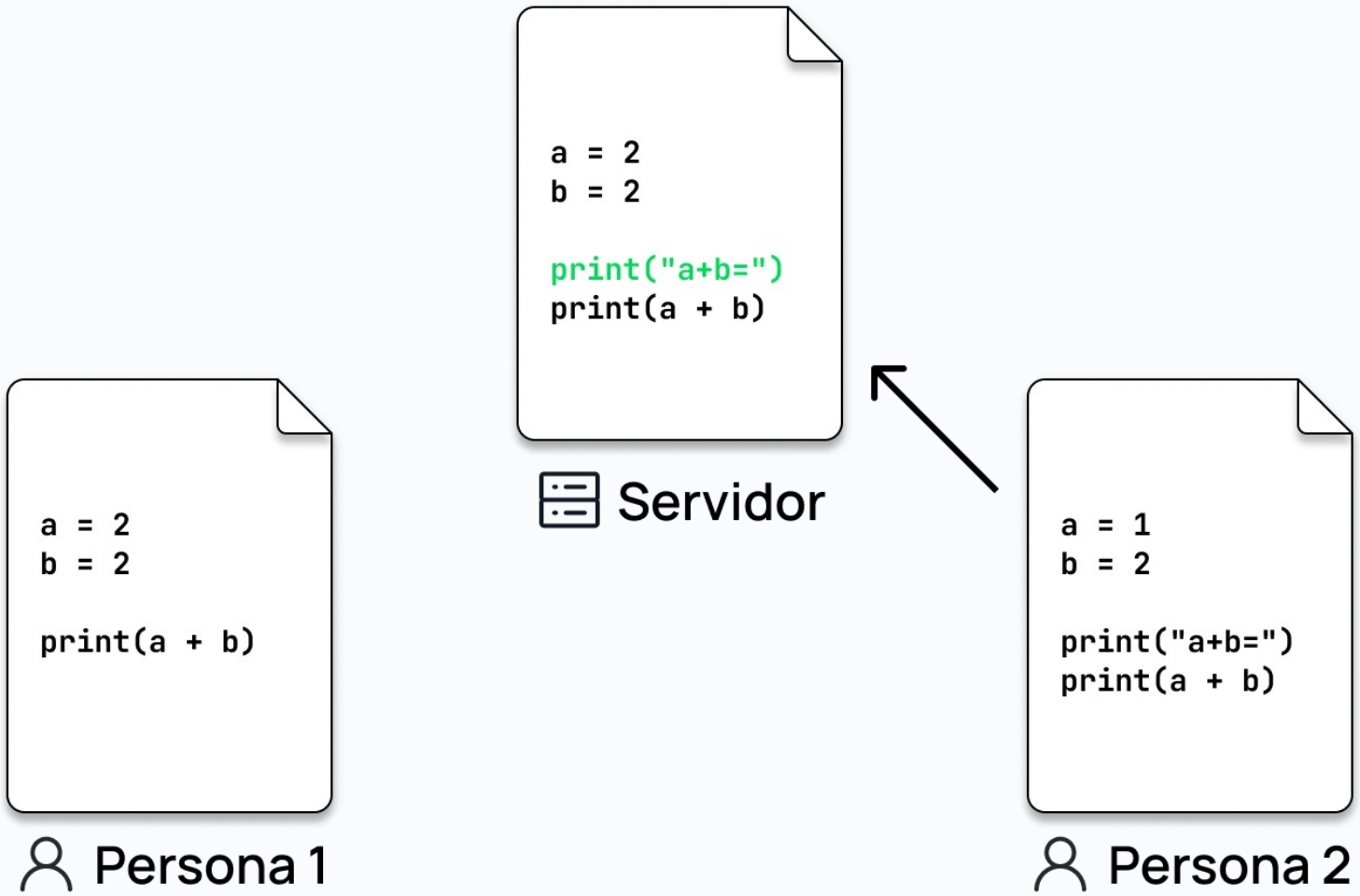
Sincronizar código entre personas



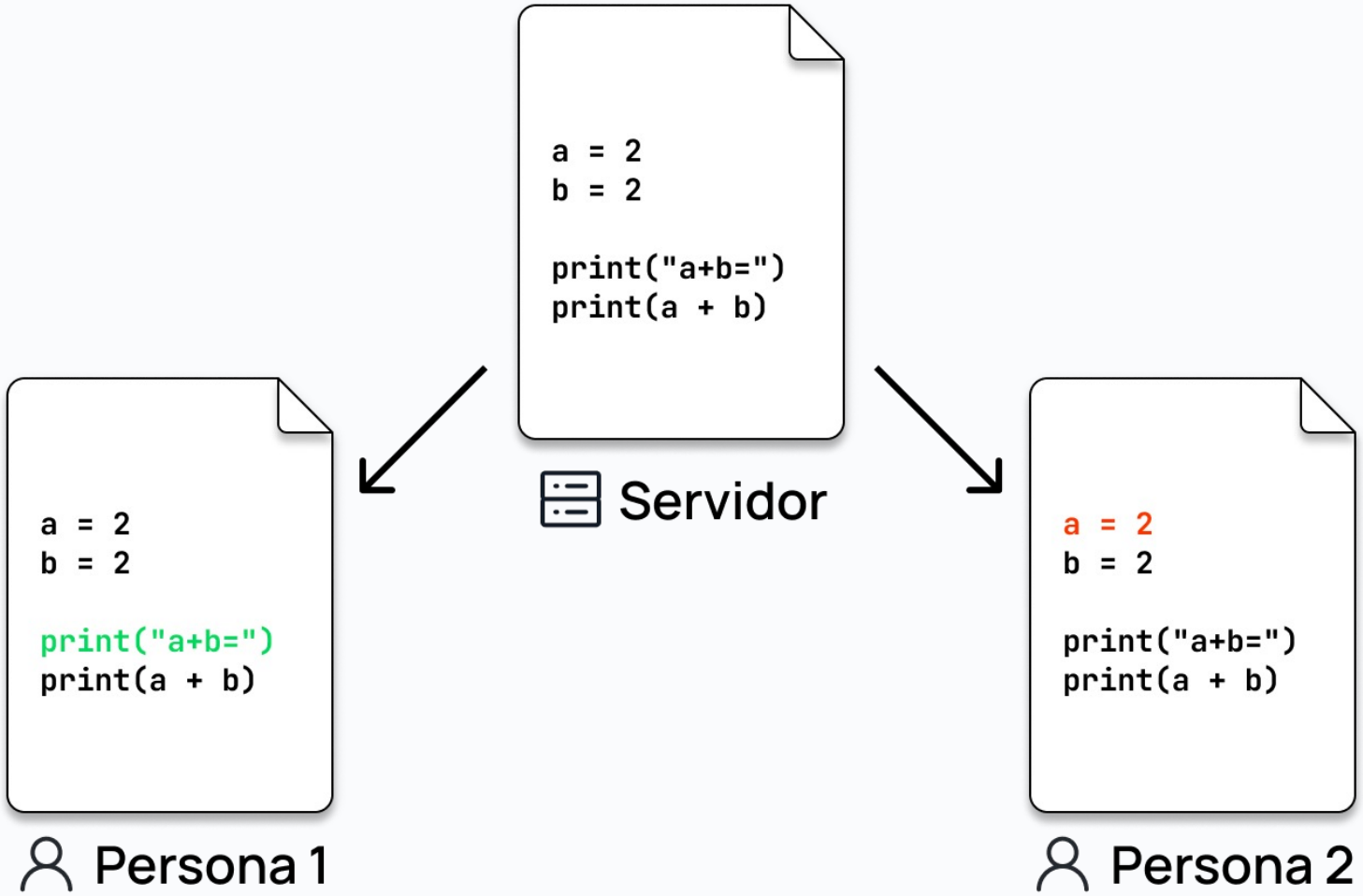
Sincronizar código entre personas



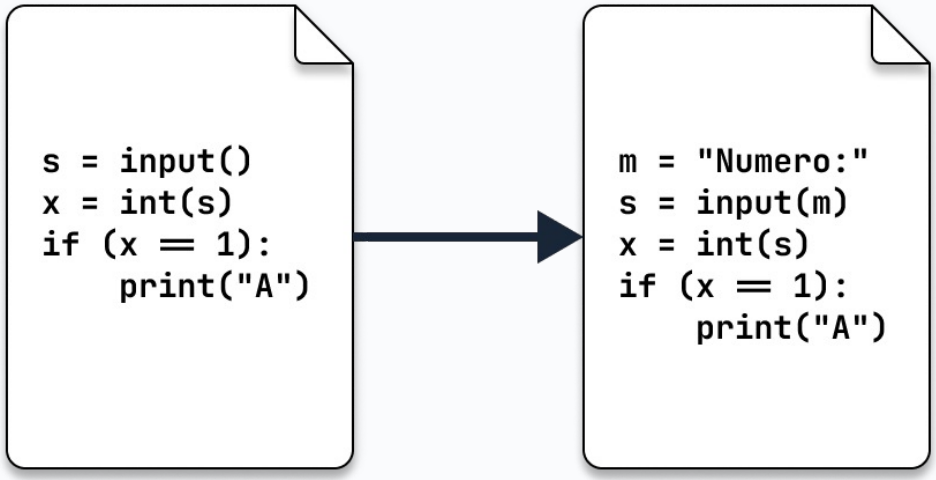
Sincronizar código entre personas



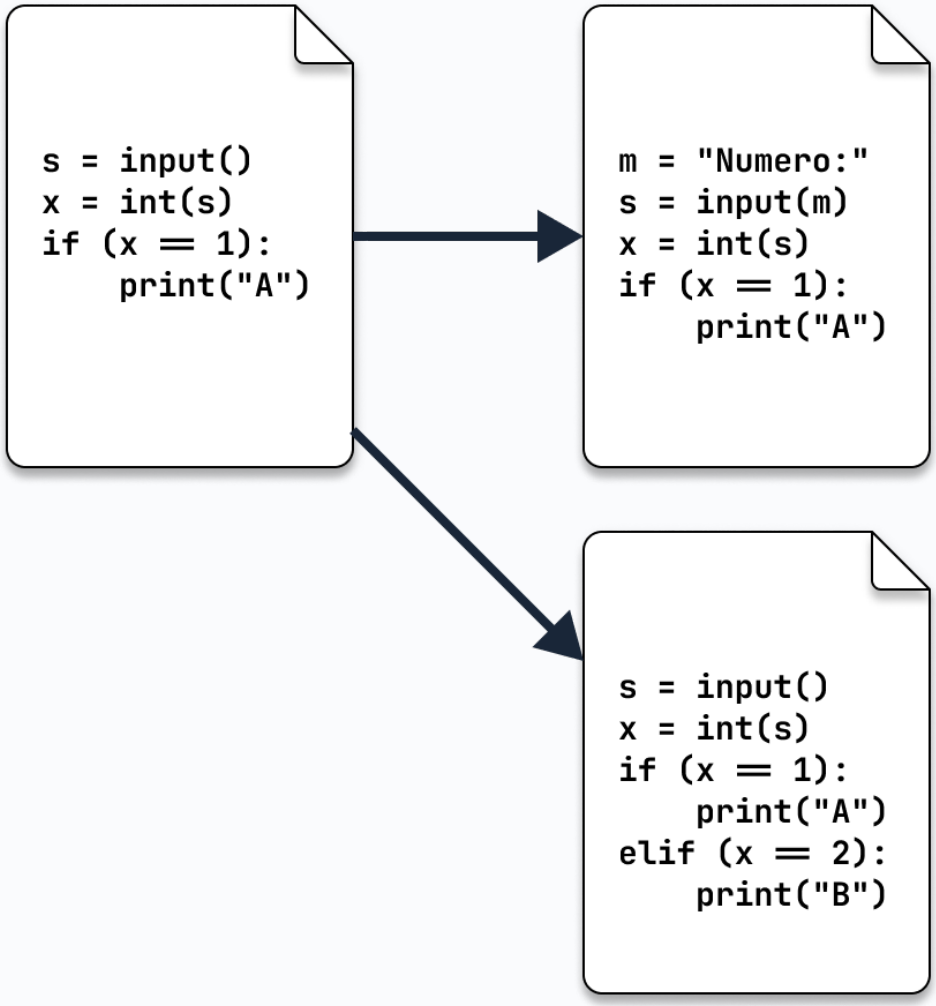
Sincronizar código entre personas



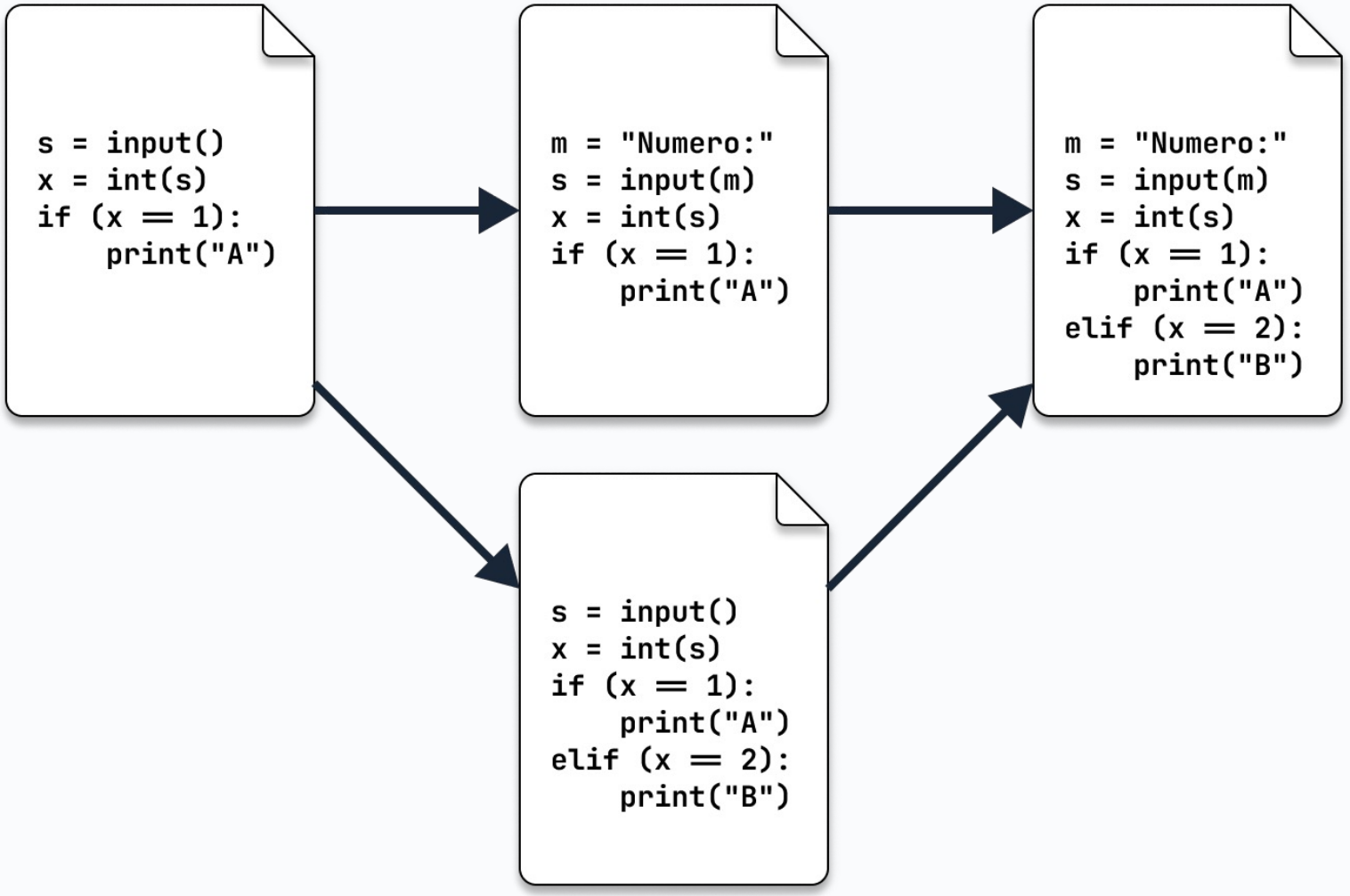
Probar cambios sin perder lo anterior



Probar cambios sin perder lo anterior



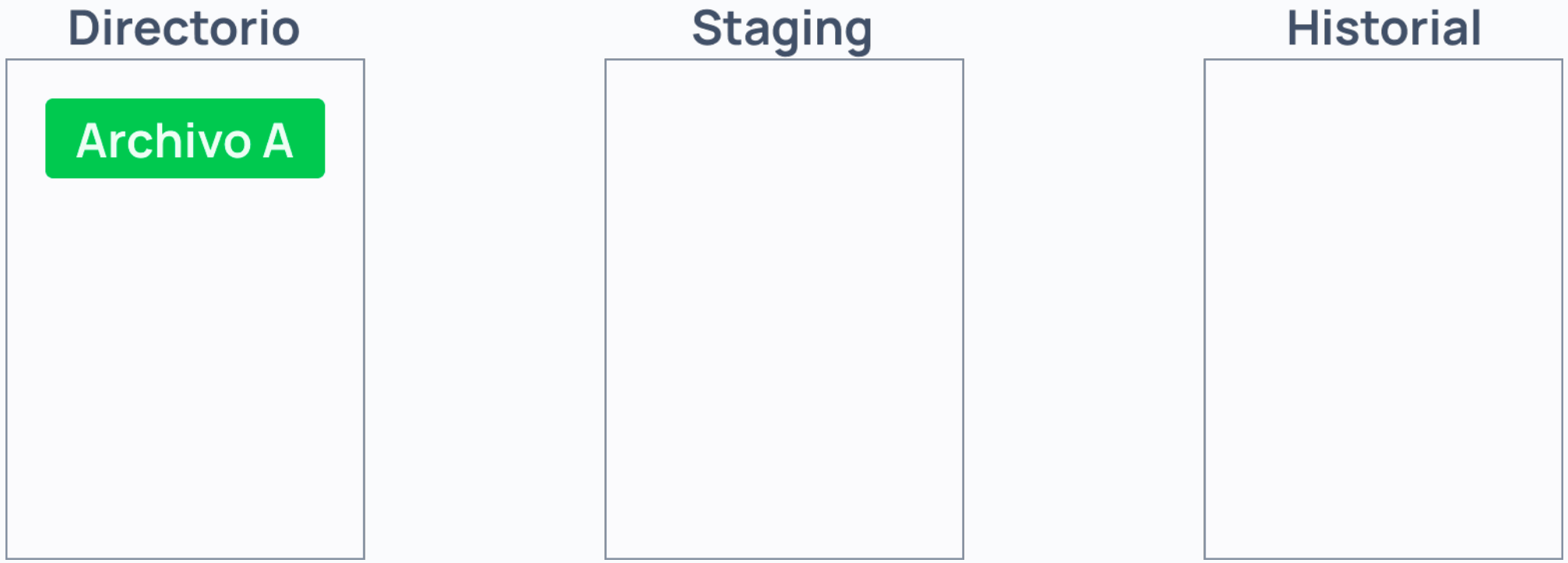
Probar cambios sin perder lo anterior



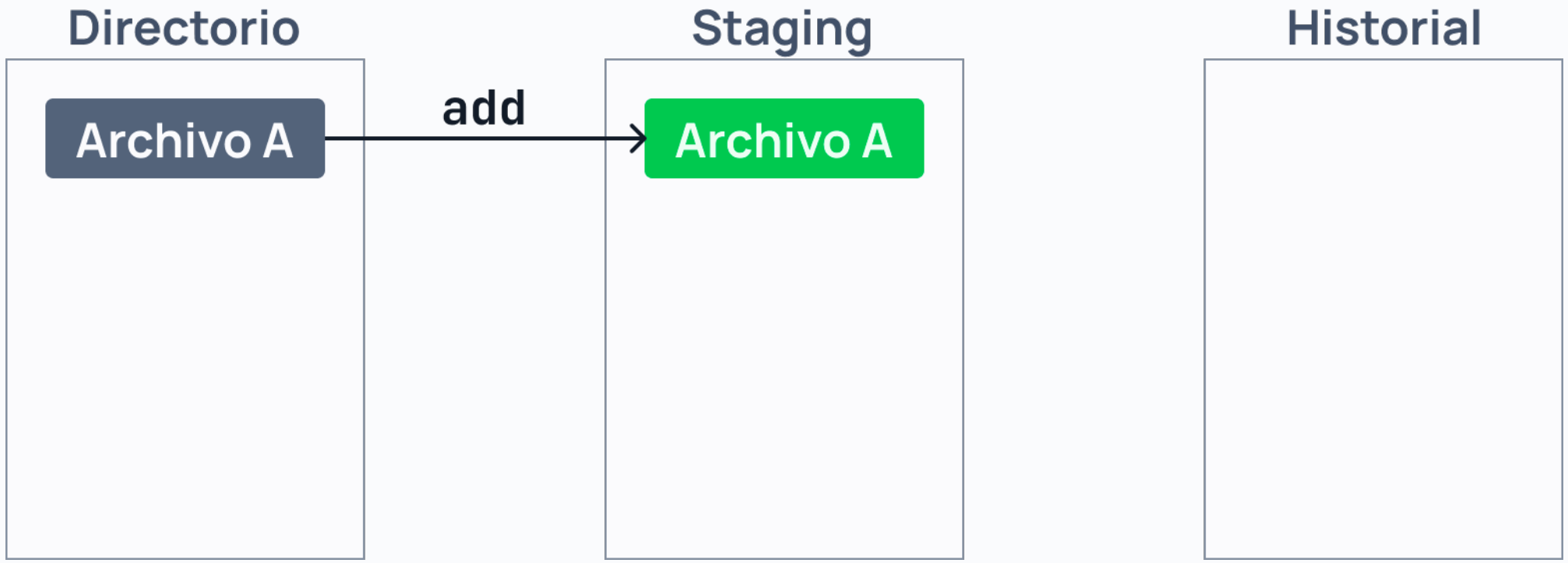
Como usar Git

`git init + git clone <url>`

git add



git add

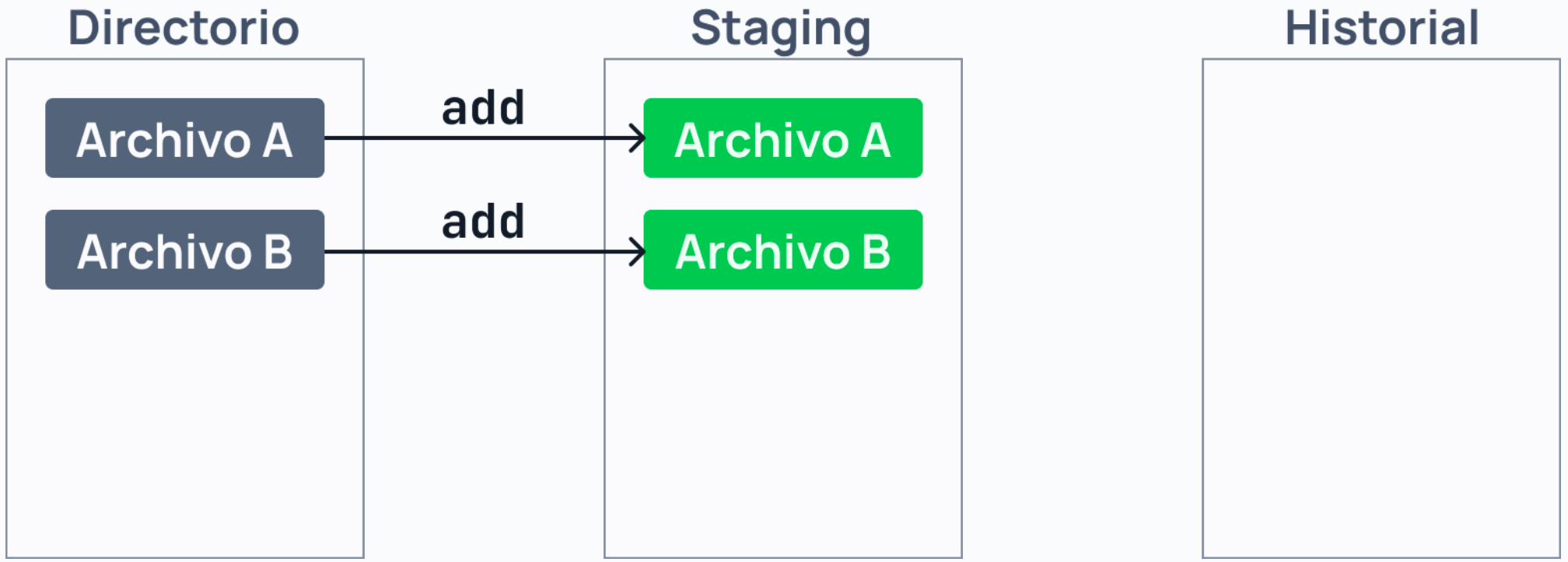


git add

git status

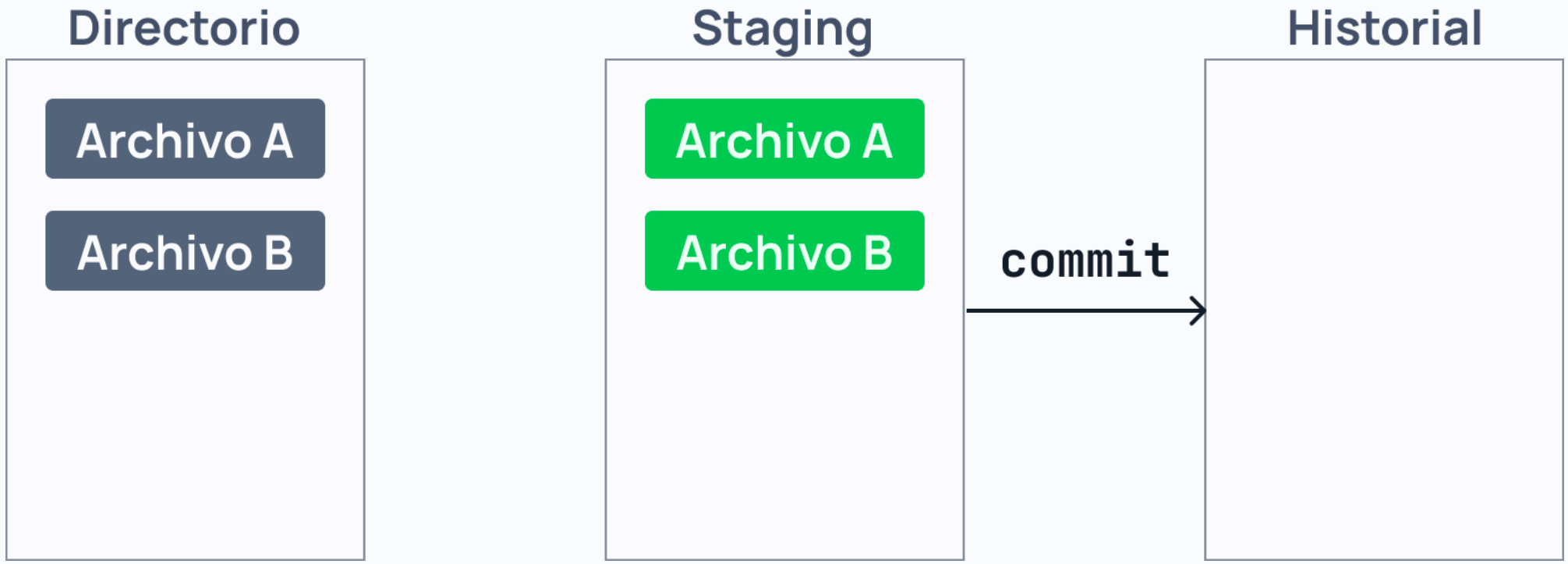

```
Changes to be committed:
        new file:   a.txt

Changes not staged for commit:
        modified:   a.txt

Untracked files:
        b.txt
```

git status

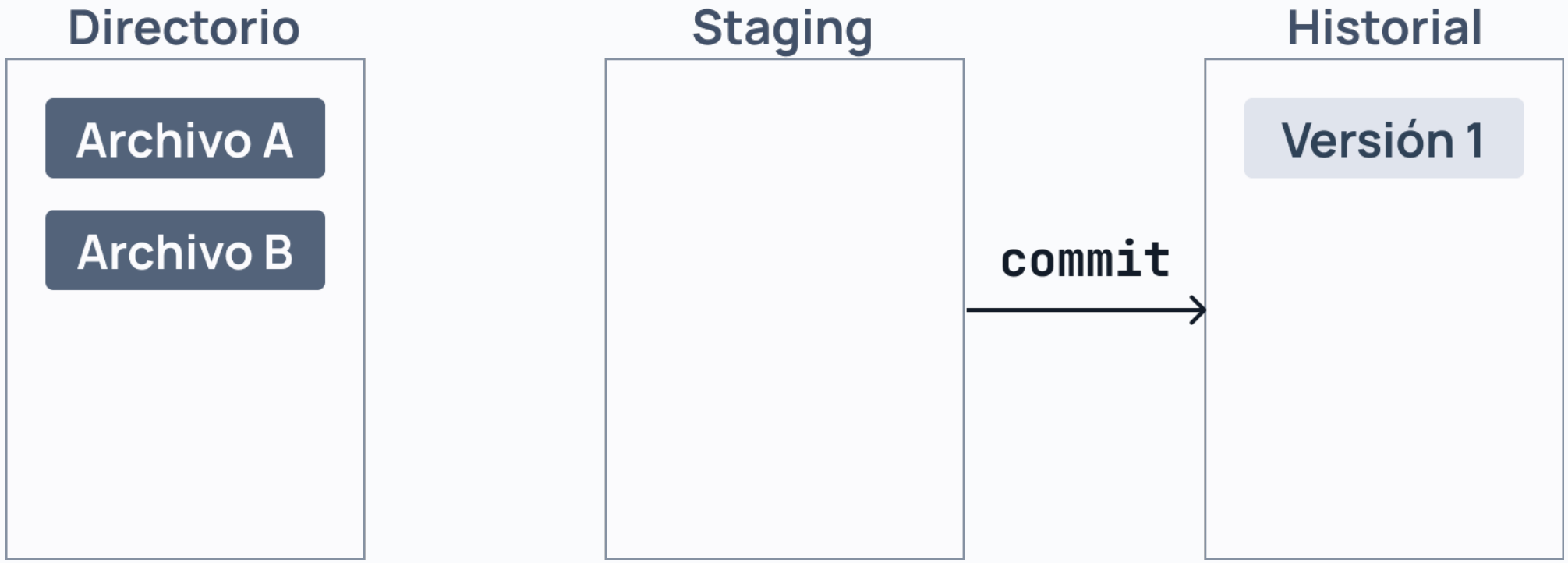



```
Changes to be committed:
        new file:   a.txt
        new file:   b.txt
```

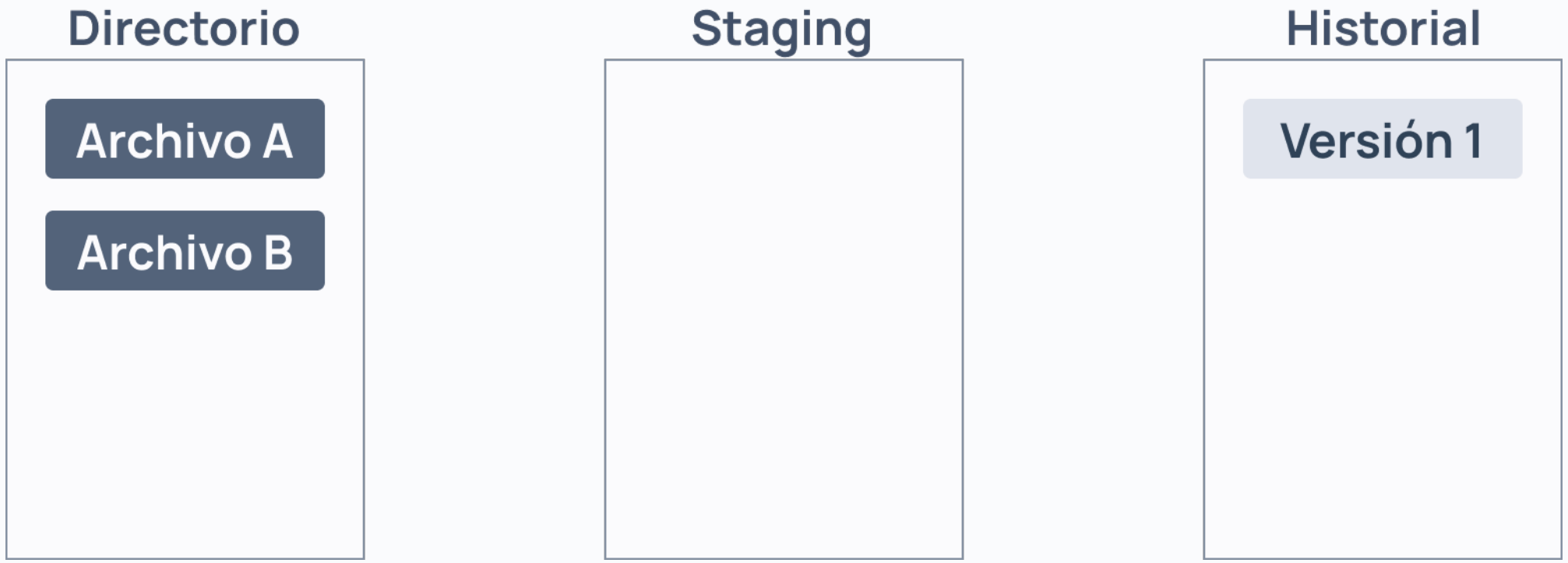
`git commit -m "version 1"`



`git commit -m "version 1"`



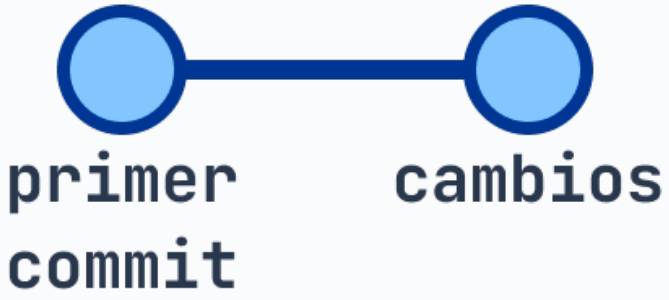
git log



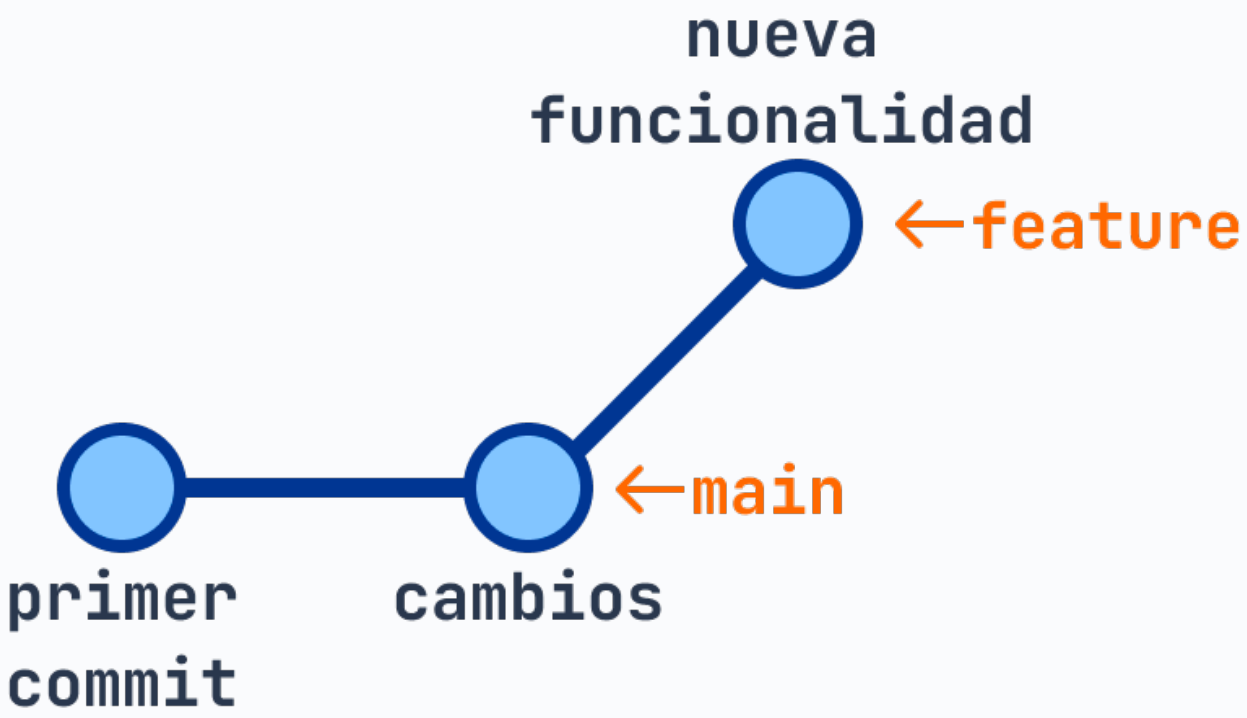

```
commit c7acbbf521f6b60408005f63c02af179a51fce64 (HEAD -> main)
Author: Benjamín Vicente <mail>
Date:   Sun Aug 20

    version 1
```

Branching

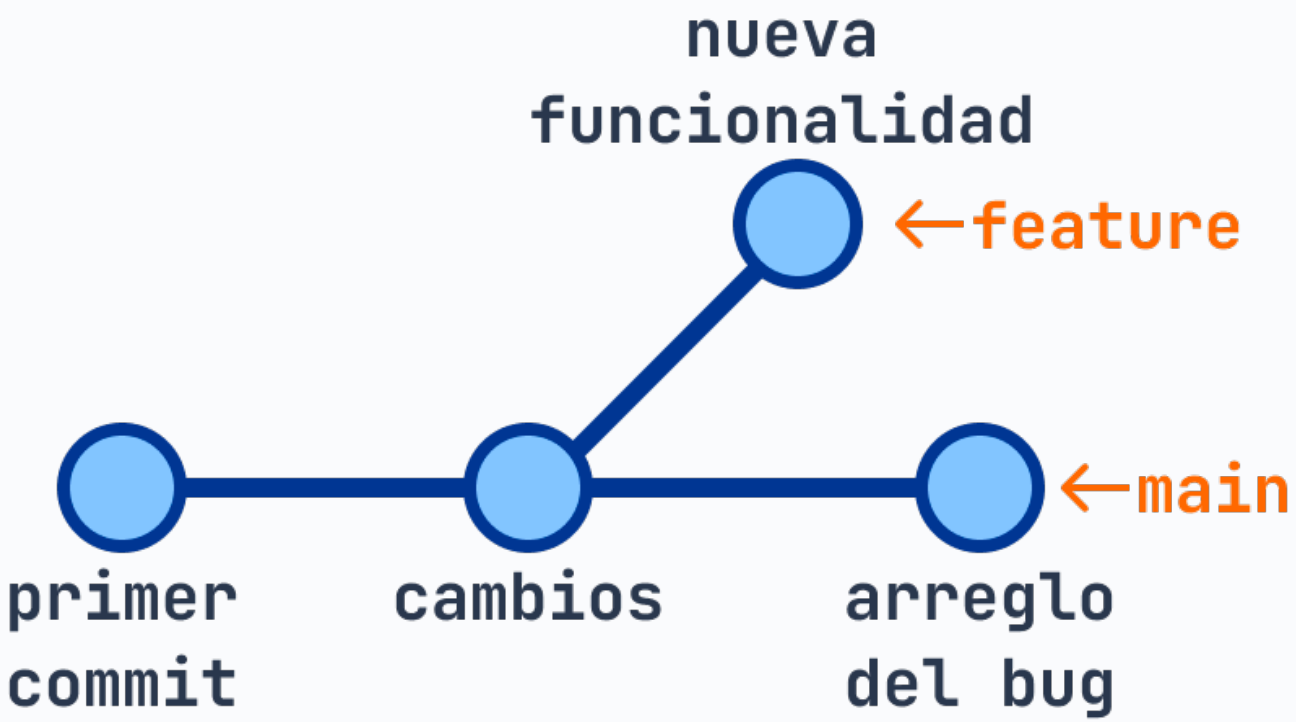
Branching



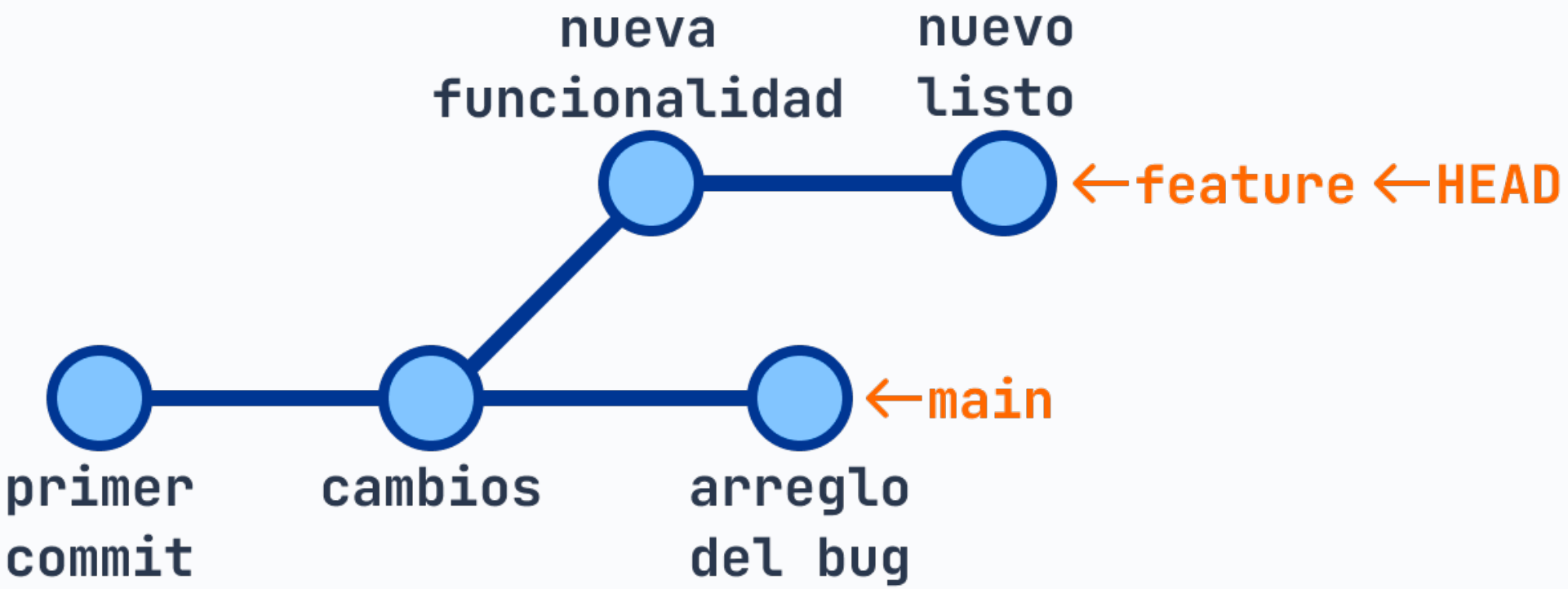
git switch -c feature



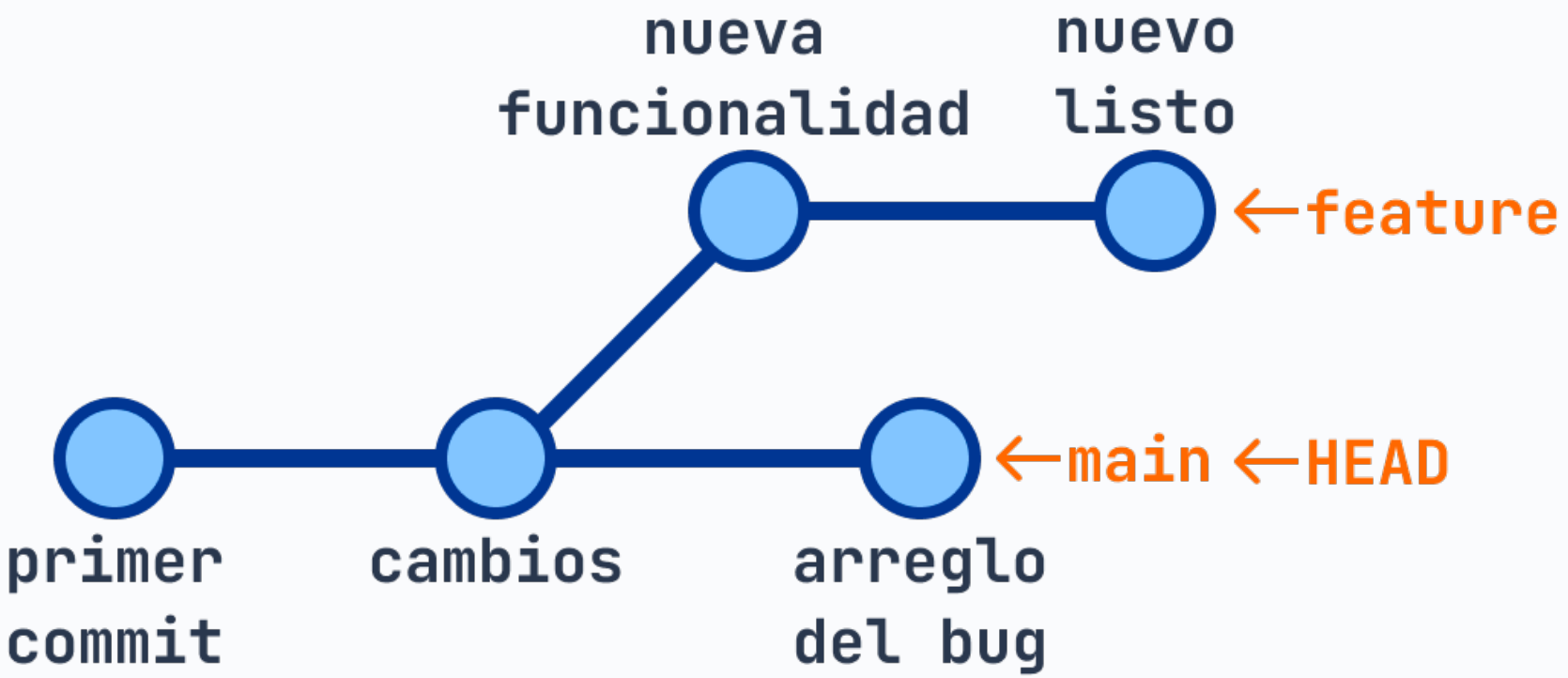
Branching



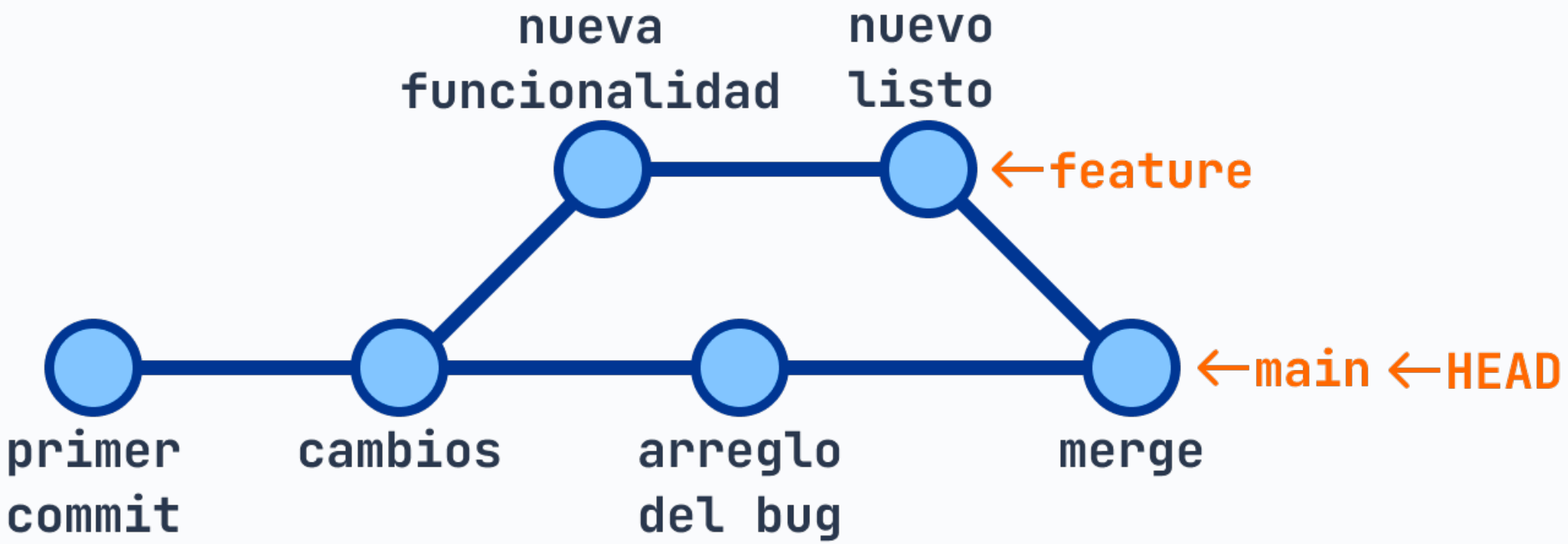
HEAD



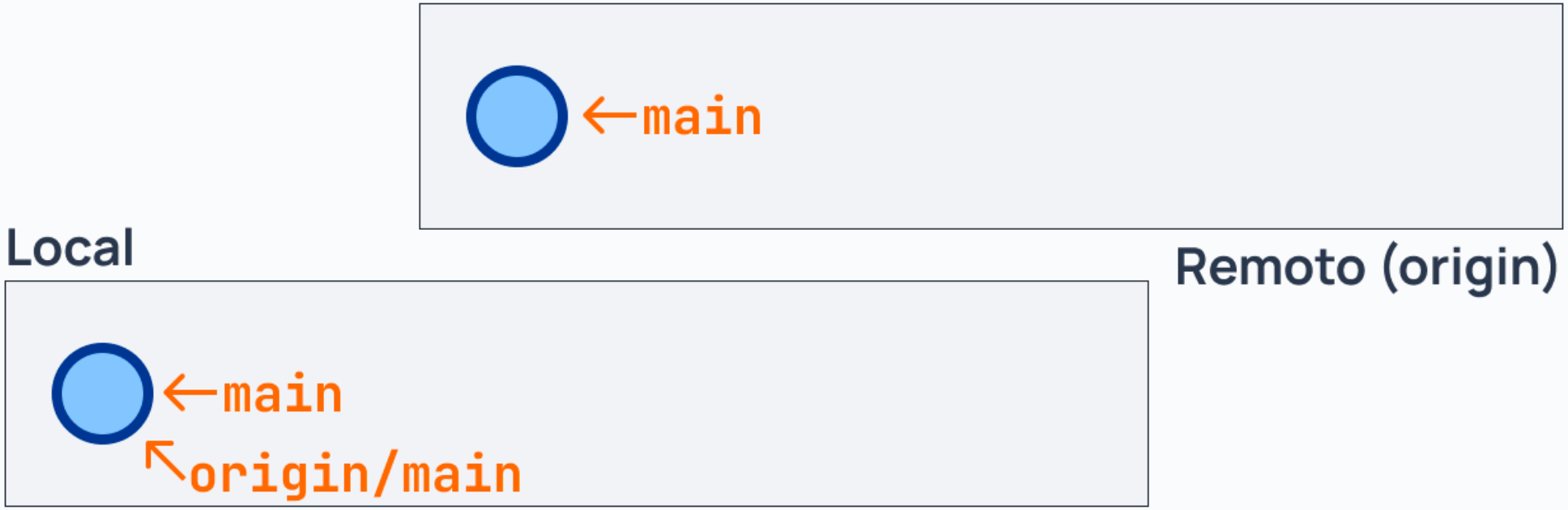
git merge feature



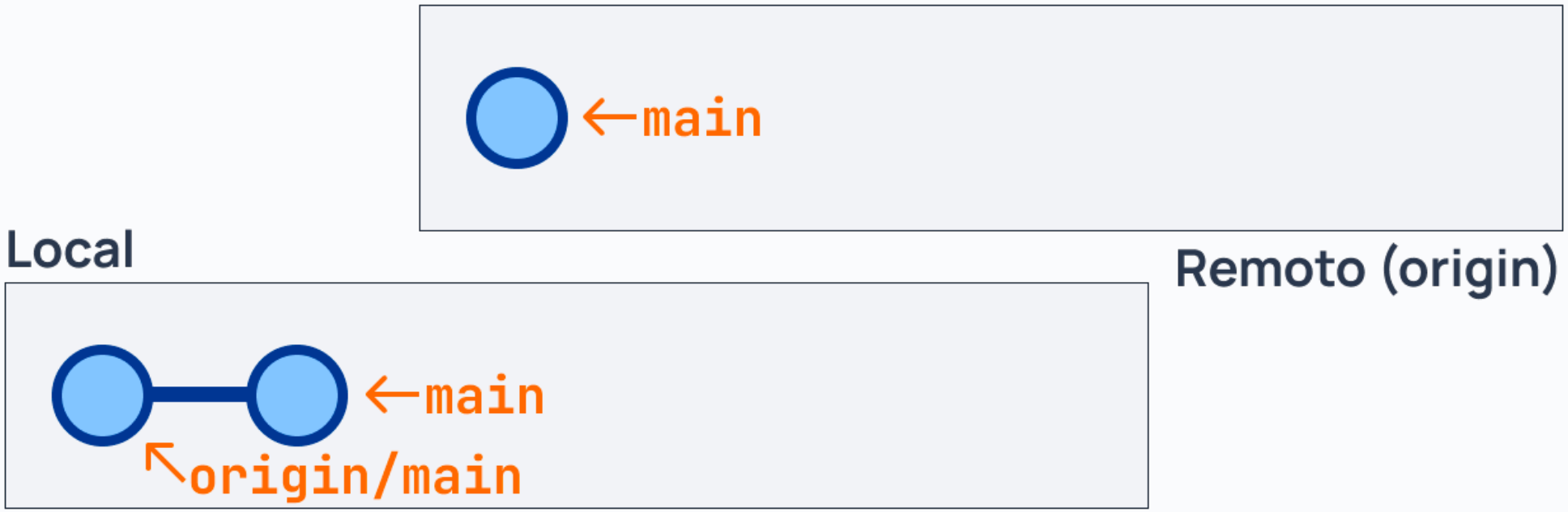
git merge feature



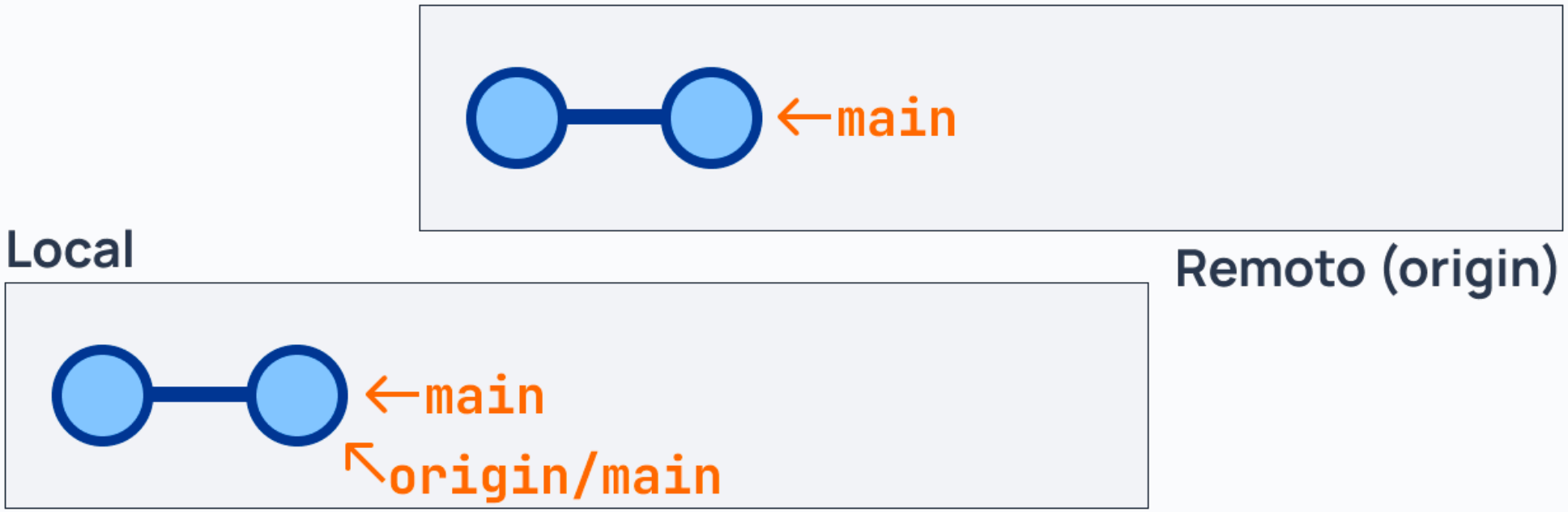
¿Que es “origin”? `git remote`



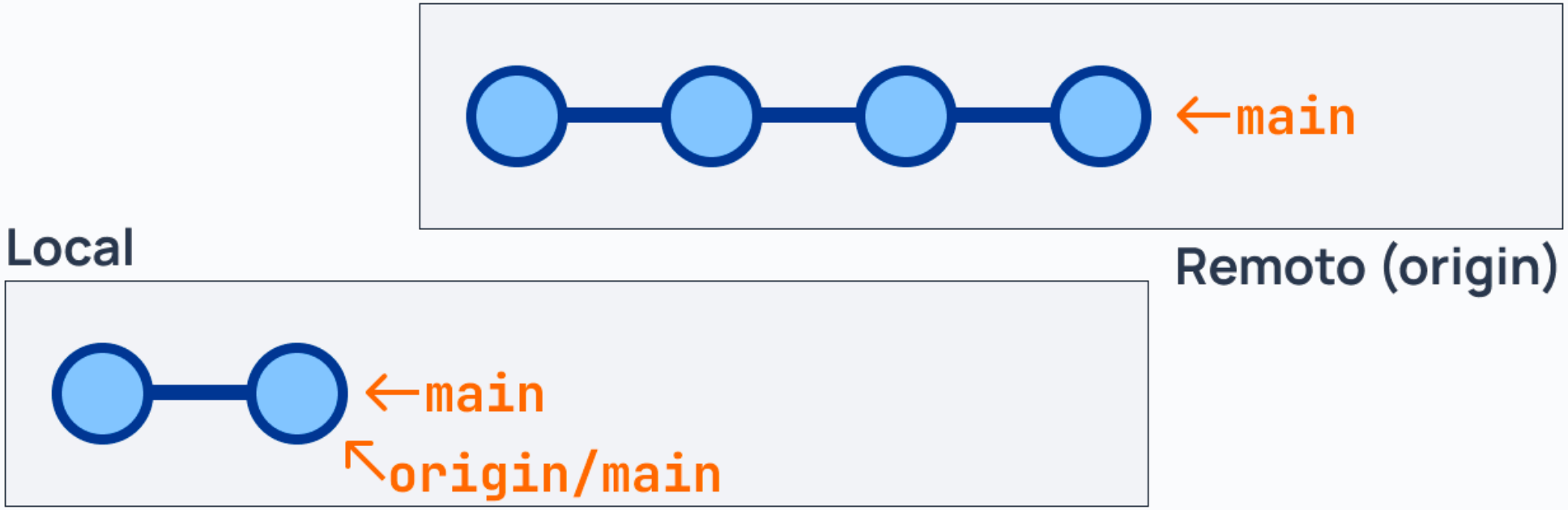
git push



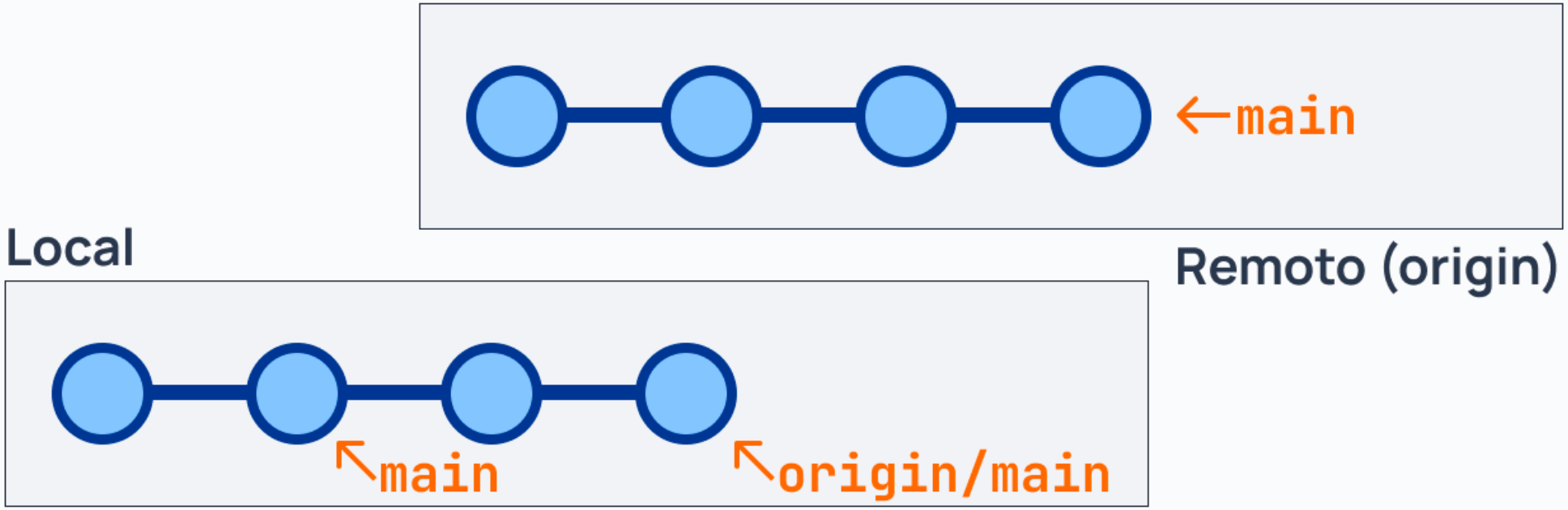
git push



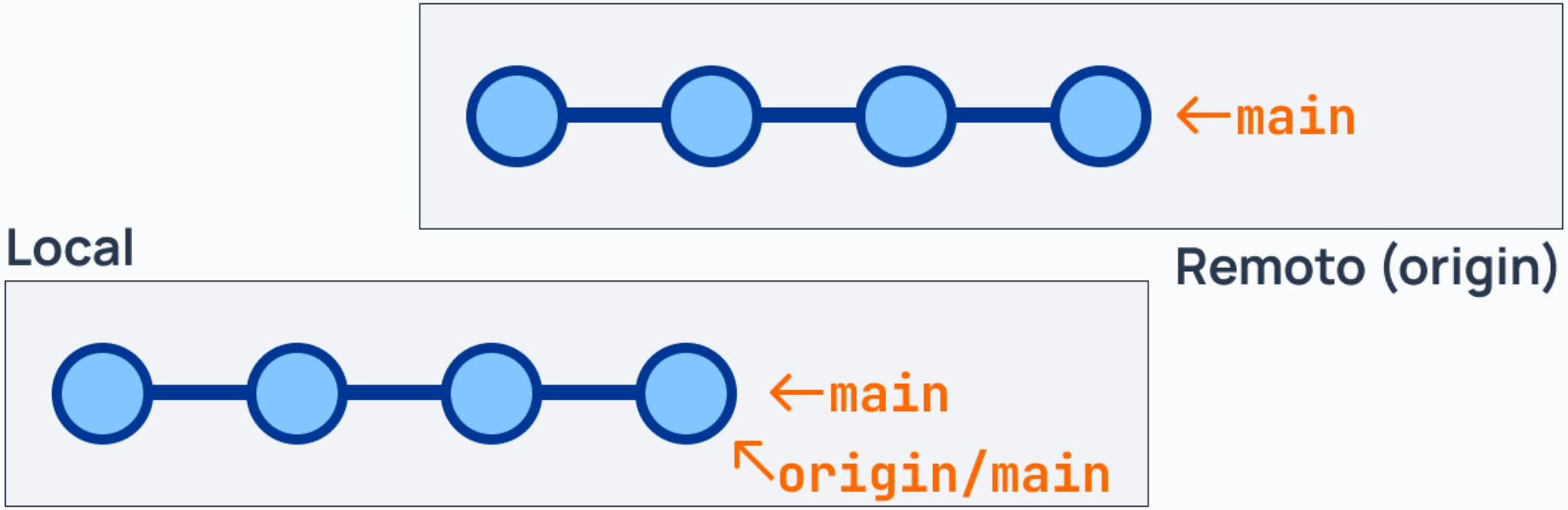
git pull = fetch + merge



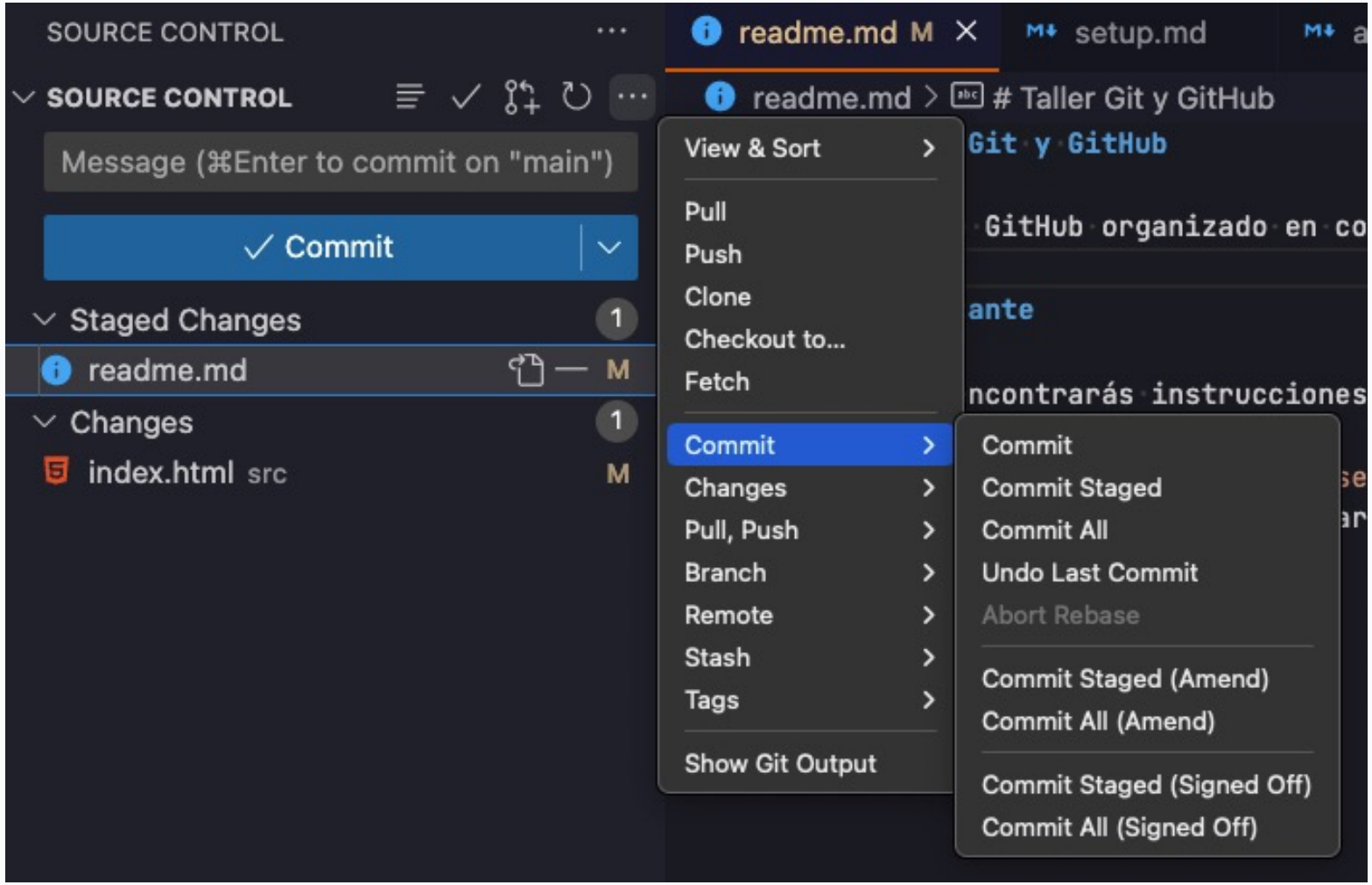
git pull = fetch + merge



git pull = fetch + merge



Git fuera del terminal



GitHub

¿Qué es lo que le falta a Git local?

- Servidor de Git
- Reporte de errores (issues)
- Discusiones de cambios (PRs)

Tour por GitHub

Actividad

Primer repo, fork y página personal 🚀

1. Crear un repositorio y clonarlo o abrirlo
2. Añade el archivo HTML
3. Añadir tu nombre a la página usando branches
4. Opcional: ¡Publica tu página!
5. Forkea el repositorio del de al lado
6. Crea una PR con confetti usando JS al repositorio forkeado

Veamos como se hace