

Errores de programadores novatos

- Aprender lenguaje antes de la lógica
- Buscar resultados rápidos
- No programar todos los días
- Ser copypasteros
- No usar control de versiones
- Preocuparse por la edad
- No planificar (ponerse a hacer código)
- Comentar los pasos obvio, no los lógicos

CONTROL DE VERSIONES



git

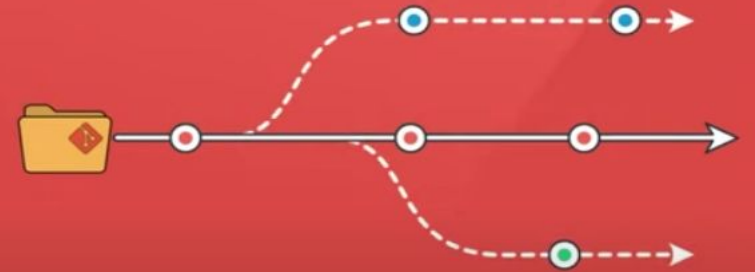
REPOSITORIO



COMMIT



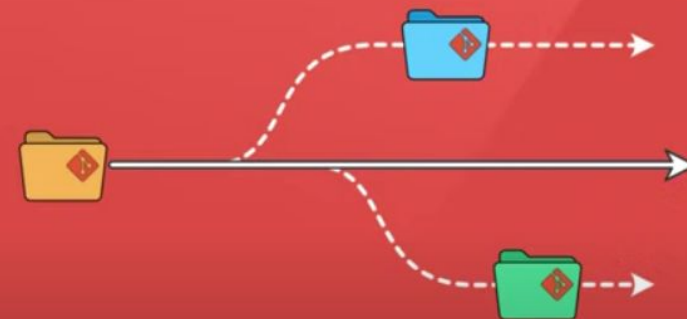
RAMAS



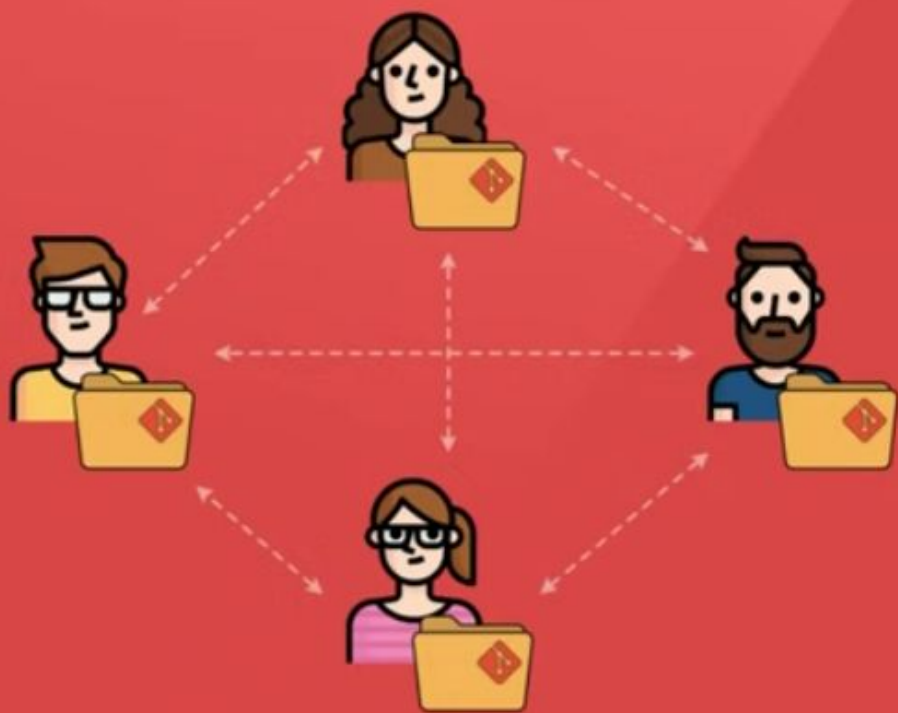
CLON



FORK



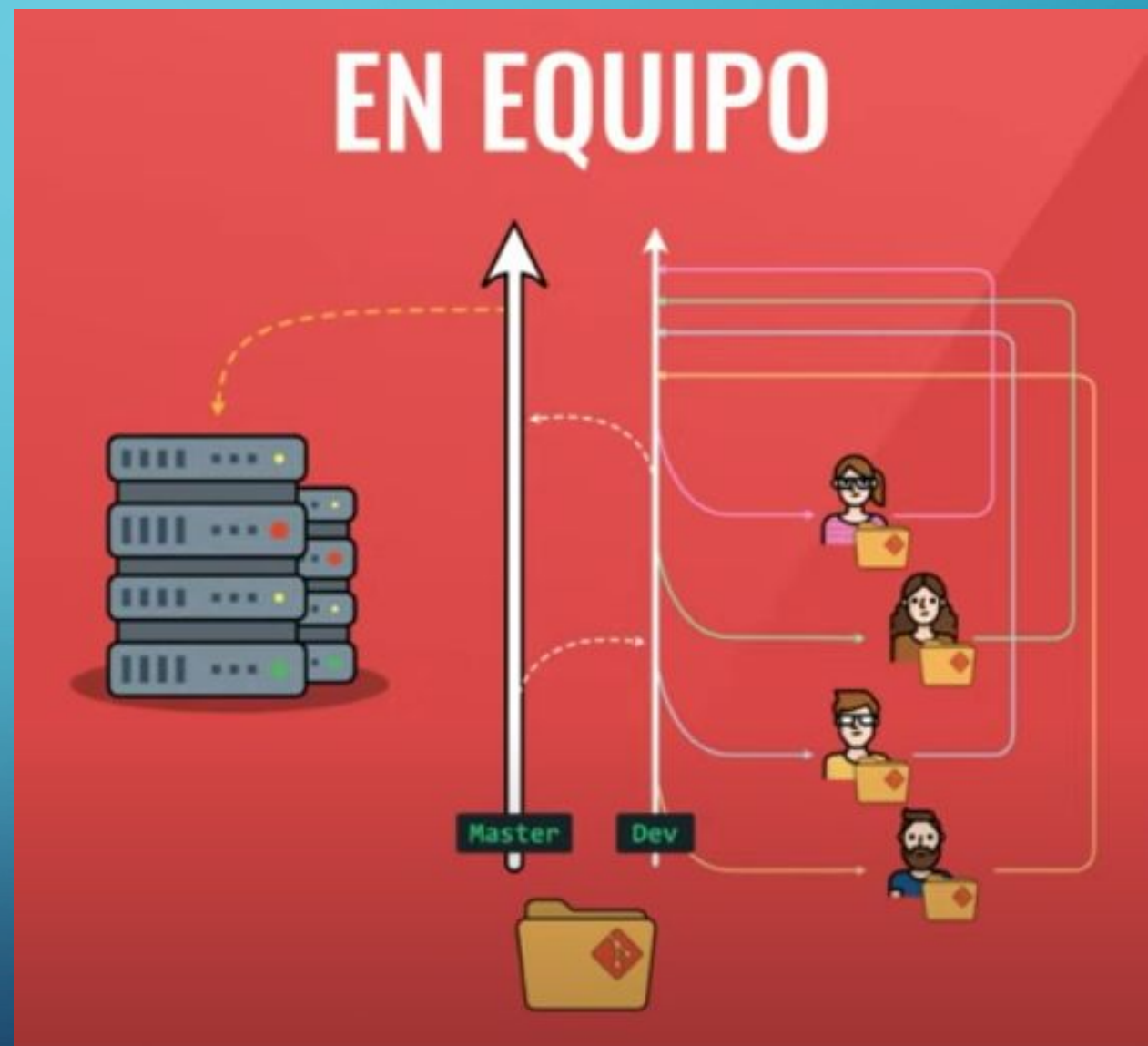
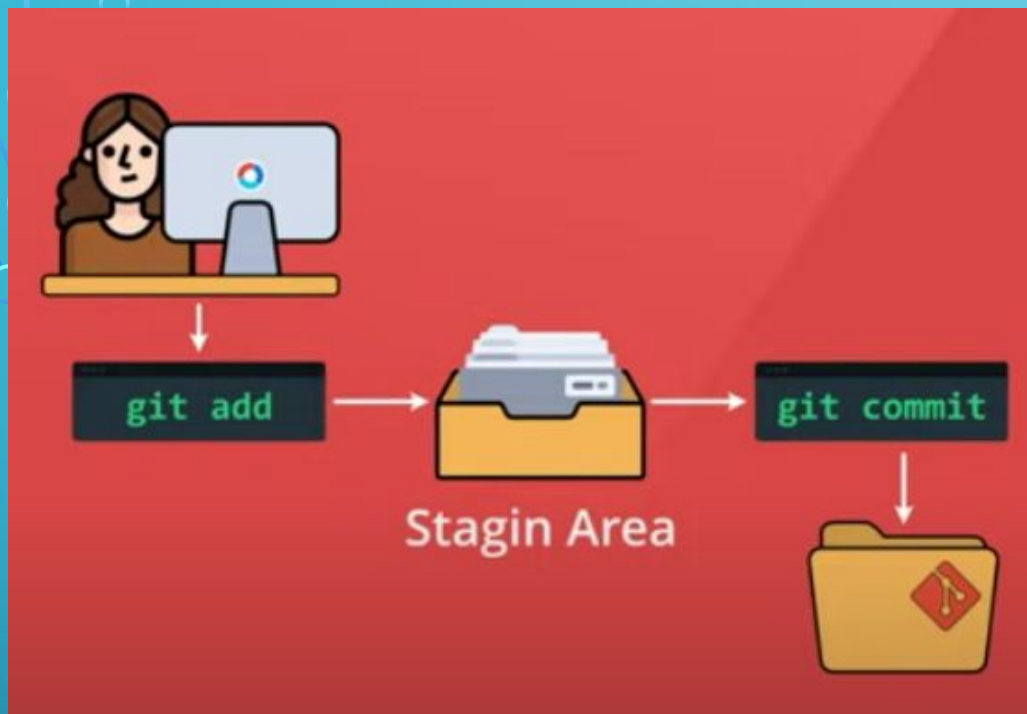
DISTRIBUIDO



RAMAS y FUSIONES

(BRANCH Y MERGE)





REPOSITARIOS EN LA NUBE



Github



Bitbucket



Gitlab

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



6. ¿Por qué nos interesa saber Git o un sistema de control de versiones?

Equipo de desarrollo



Proyecto



Equipo de desarrollo



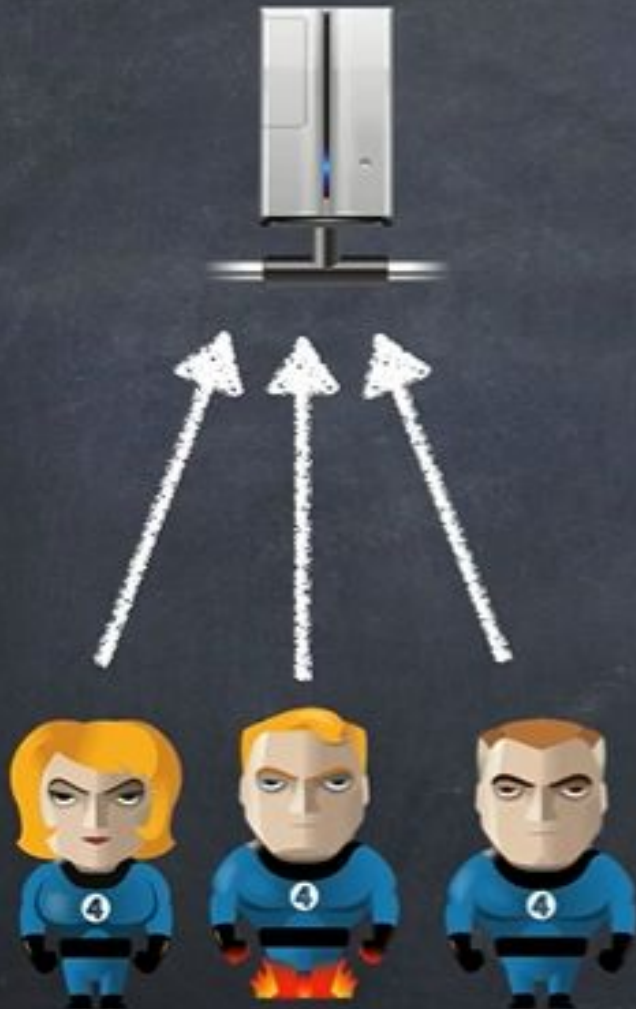
Proyecto

¿Qué hicieron?
Ayer funcionaba

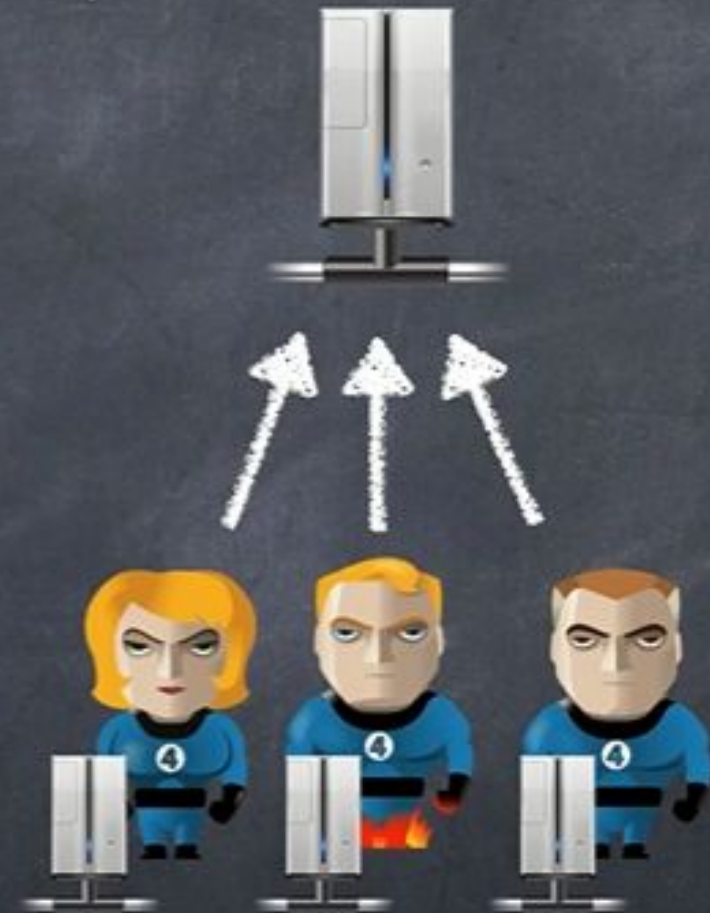


Los jefes y el destino

Repositorio Central



Repositorio distribuido



Proyecto = Repositorio

Ejemplo



¿Cómo funciona Git?

Línea del tiempo



index.html



¿Cómo funciona Git?



día 1



día 2



día 3

Línea del tiempo

index.html



Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

[https://github.com/LauraAchetta/pr](https://github.com/LauraAchetta/prueba.git)

et started by [creating a new file](#) or [uploading an existing file](#). We recommend every re

..or create a new repository on the command line

```
echo "# prueba" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/LauraAchetta/prueba.git
git push -u origin main
```

..or push an existing repository from the command line

```
git remote add origin https://github.com/LauraAchetta/prueba.git
git branch -M main
git push -u origin main
```

C:\Windows\System32\cmd.exe

```
C:\argentin\MLA(SI)\Clase desarrollo>git init
Initialized empty Git repository in C:/argentin/MLA(SI)/Clase desarrollo/.git/

C:\argentin\MLA(SI)\Clase desarrollo>git remote add origin https://github.com/LauraAchetta/prueba.git

C:\argentin\MLA(SI)\Clase desarrollo>git add .

C:\argentin\MLA(SI)\Clase desarrollo>git commit -m "mi primer repositorio desde la clase"
[main (root-commit) 4b2e211] mi primer repositorio desde la clase
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Ejercicios1/CVLauraAchetta.doc

C:\argentin\MLA(SI)\Clase desarrollo>git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 16.94 KiB | 2.12 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/LauraAchetta/prueba.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\argentin\MLA(SI)\Clase desarrollo>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

C:\argentin\MLA(SI)\Clase desarrollo>
```