



git

Version Control System



Git

Version Control System

Apache Subversion

Software

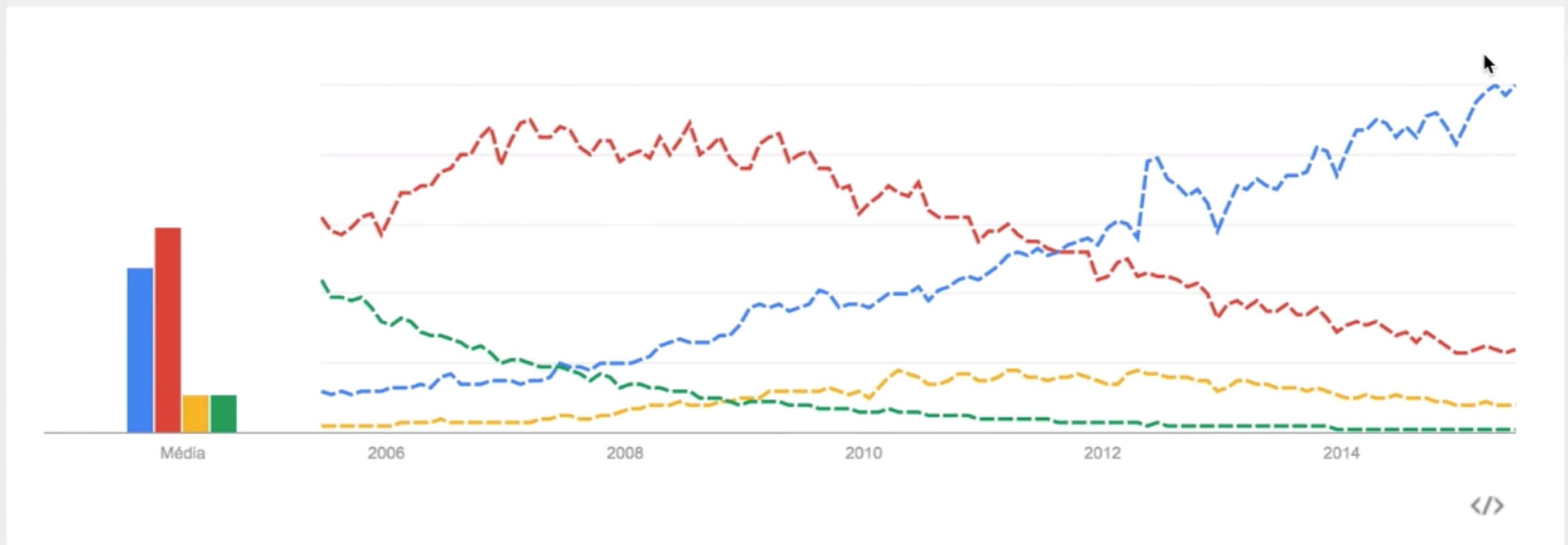
Mercurial

Revision Control Software

Concurrent Versions ...

System software

Interesse com o passar do tempo



VI. SOURCE CONTROL

Overview

Developer Profile

Technology

I. Most Popular Technologies

II. Most Loved, Dreaded, and Wanted Tools

III. Desktop Operating System

IV. Text Editor

V. IDE Theme

VI. Source Control

VII. Tabs vs. Spaces

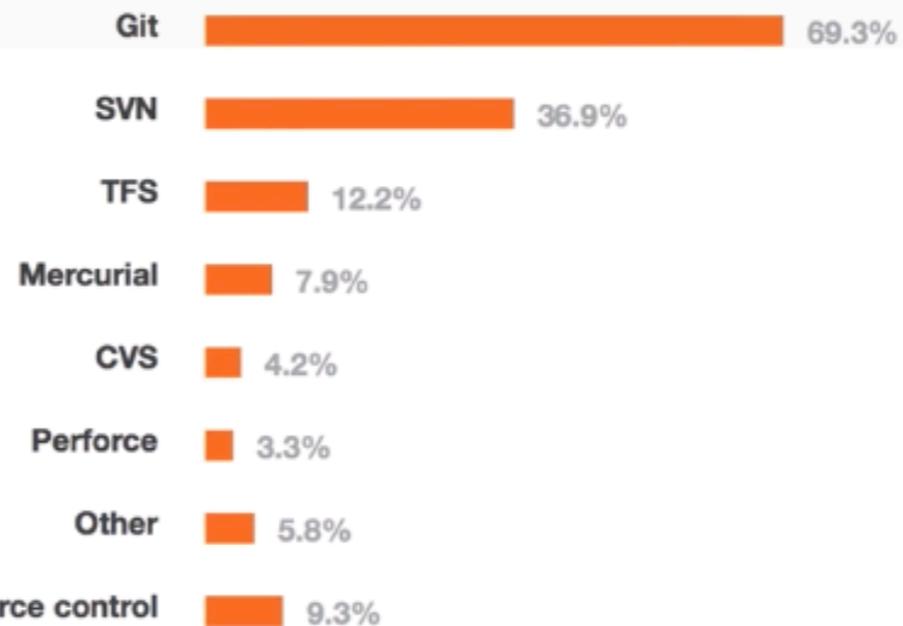
VIII. Caffeine

Work

Community

Back to top 

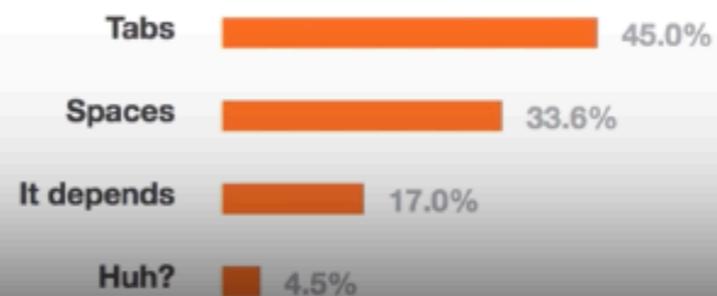
[Get a better job](#)

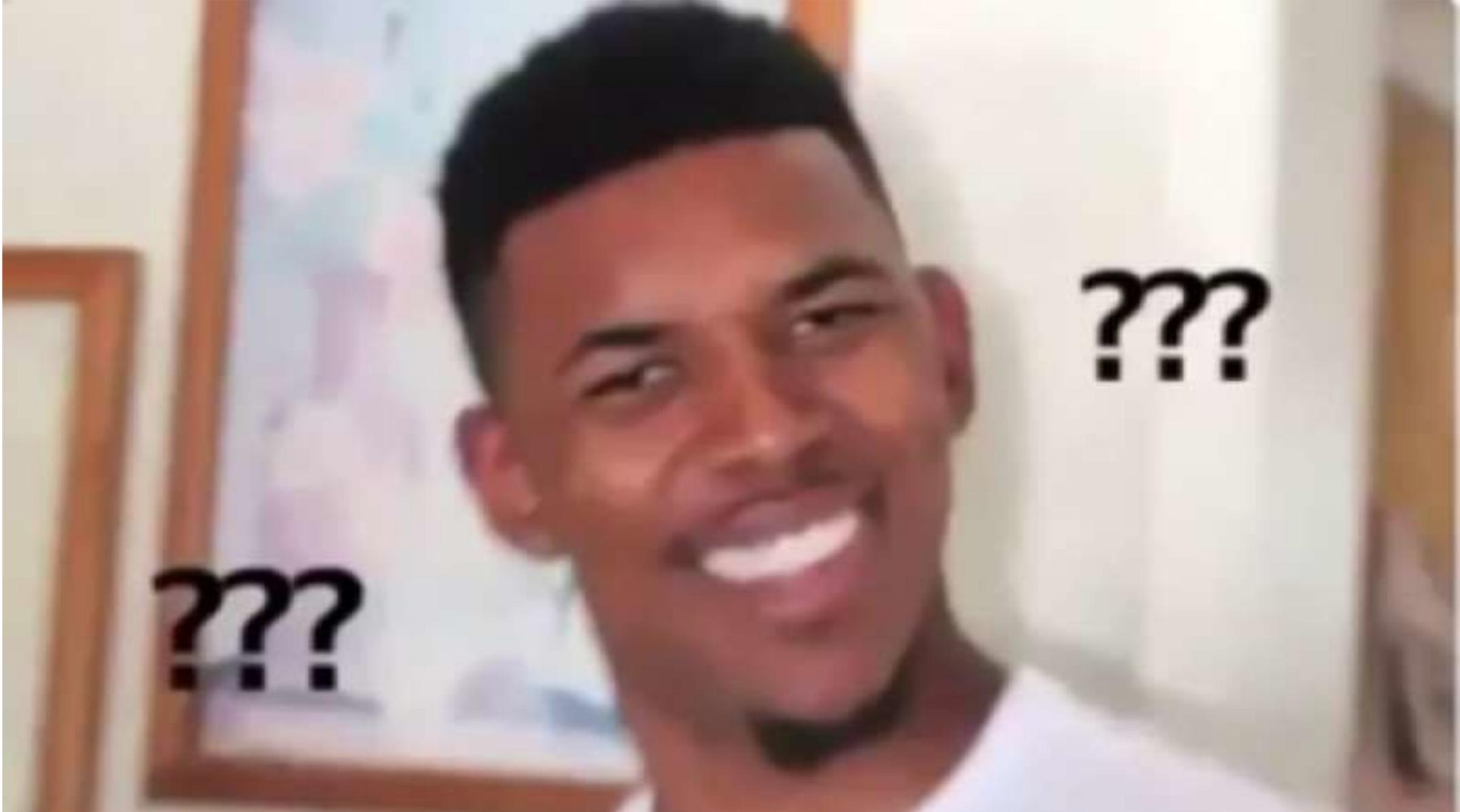


16,694 responses

Source control is an industry standard tool. However, fun fact: about 10% of developers still don't use it.

VII. TABS VS. SPACES

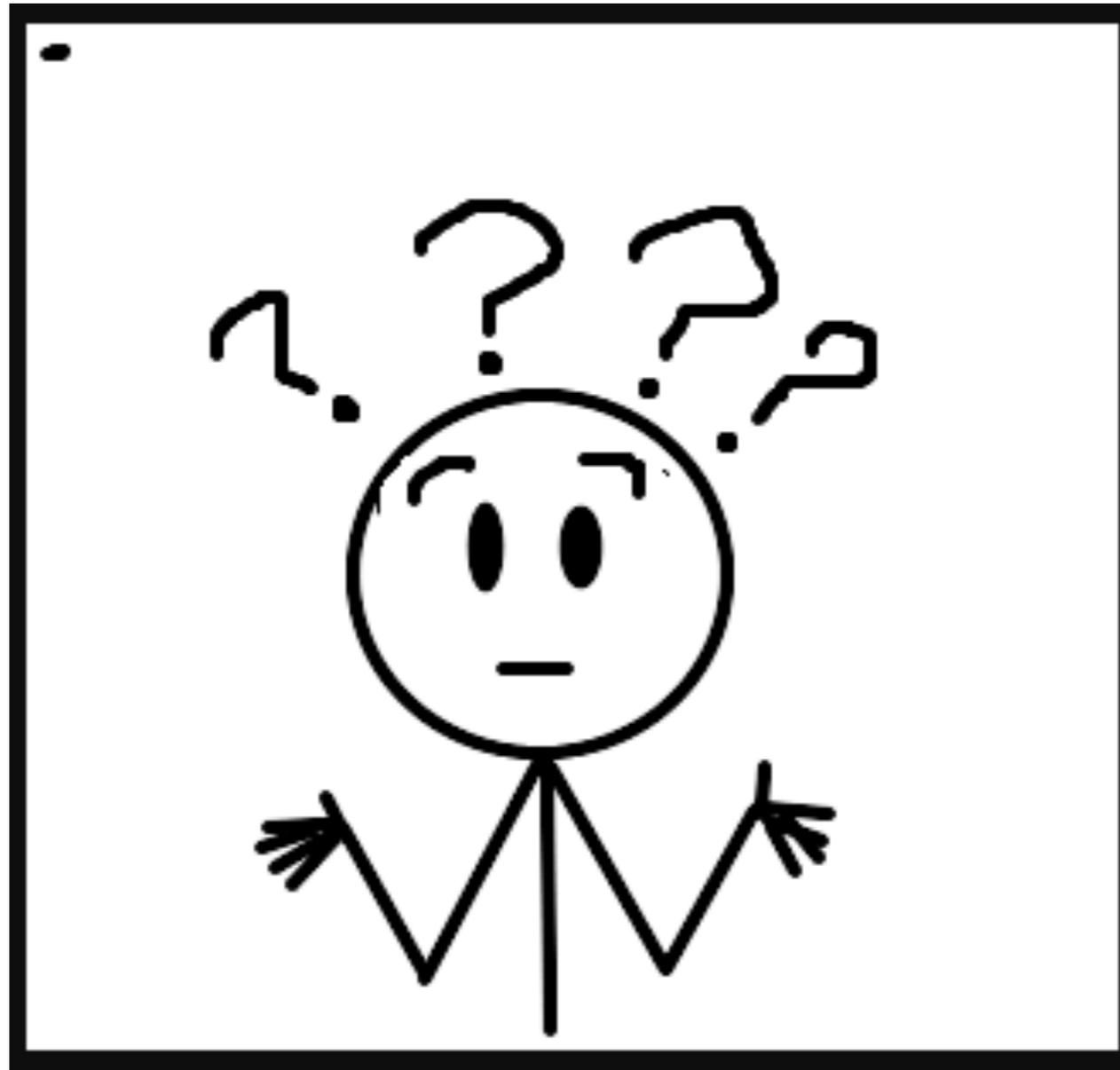




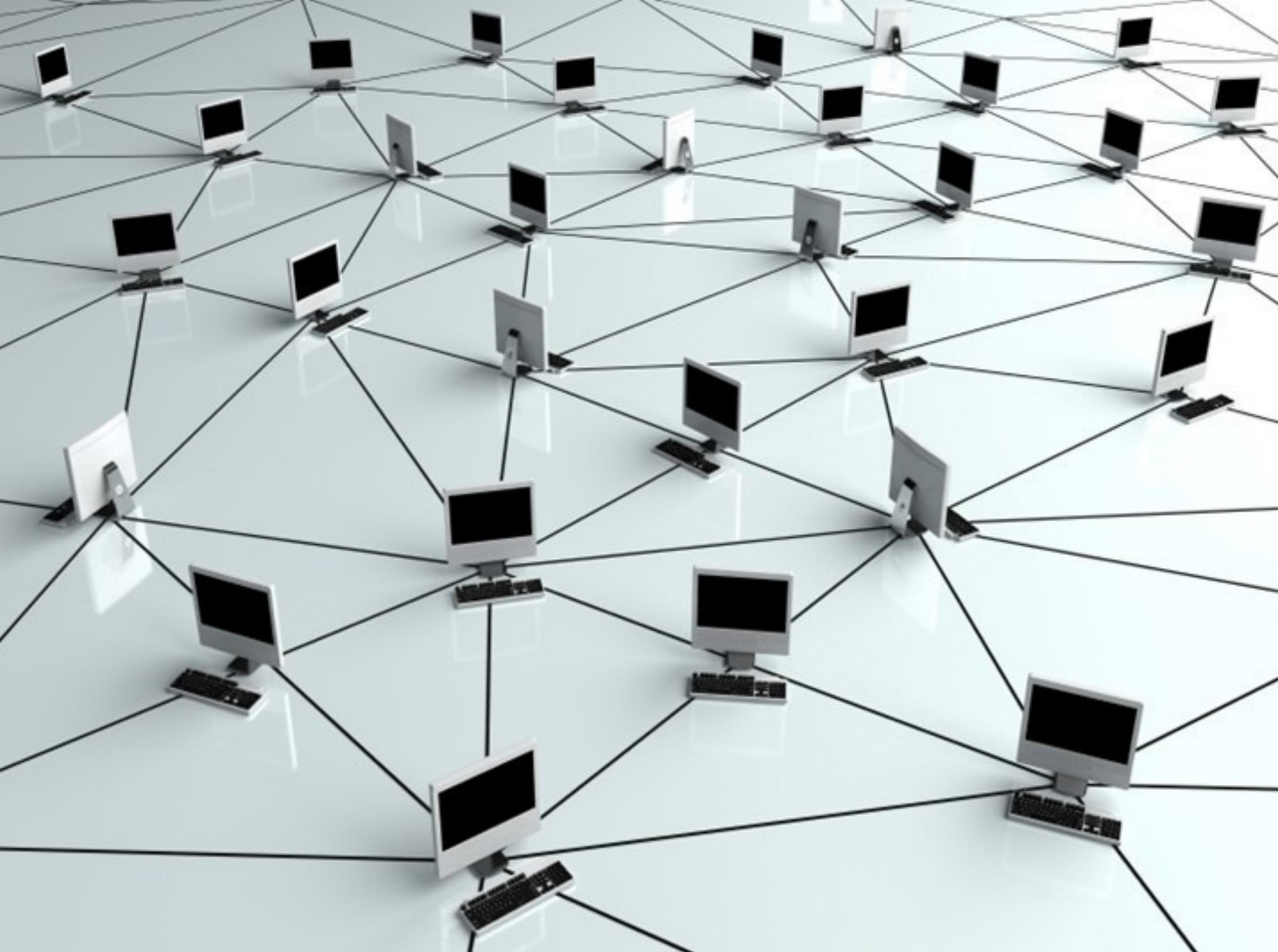
Histórico

Trabalhar simultaneamente

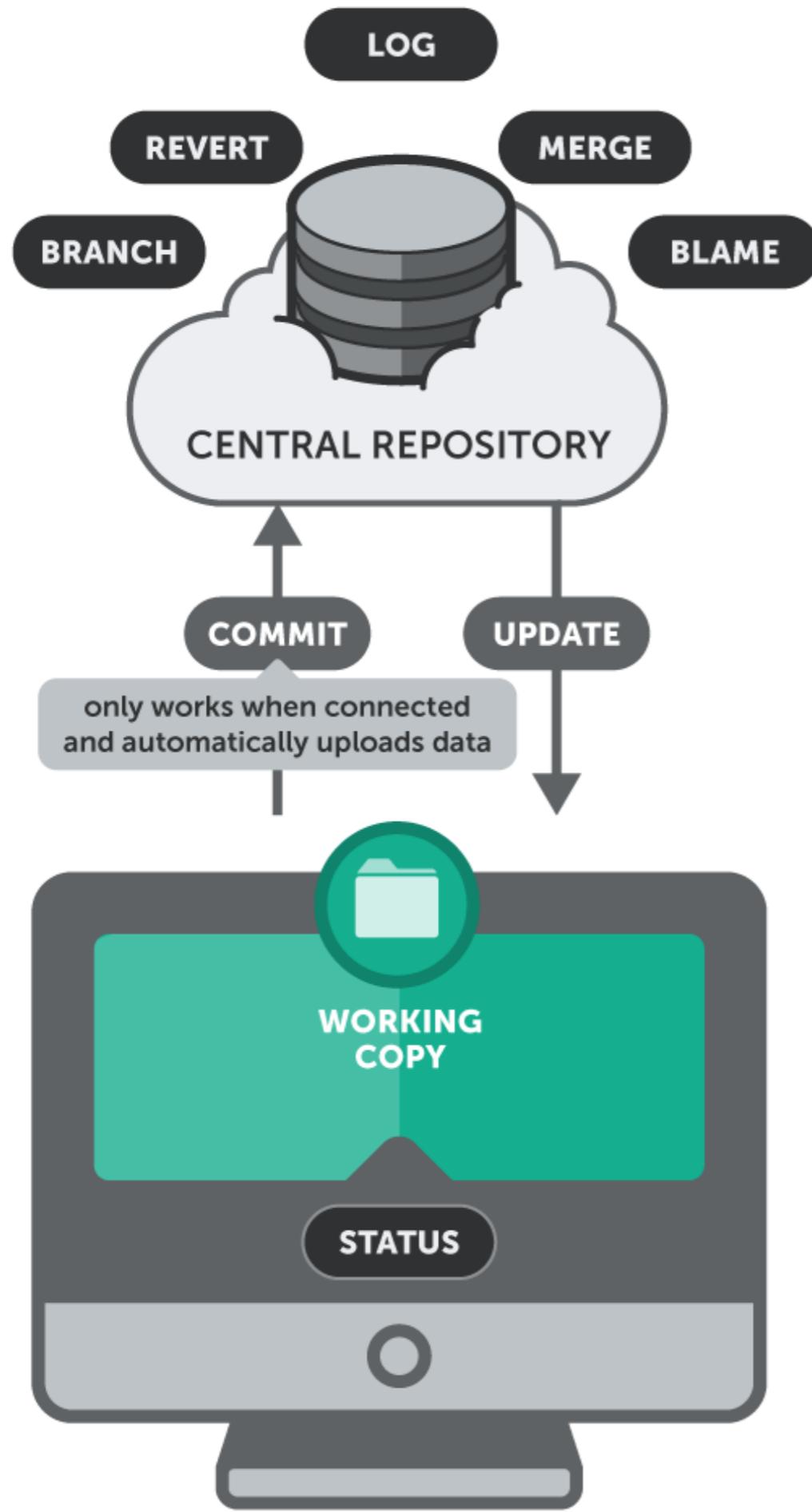
Versionar



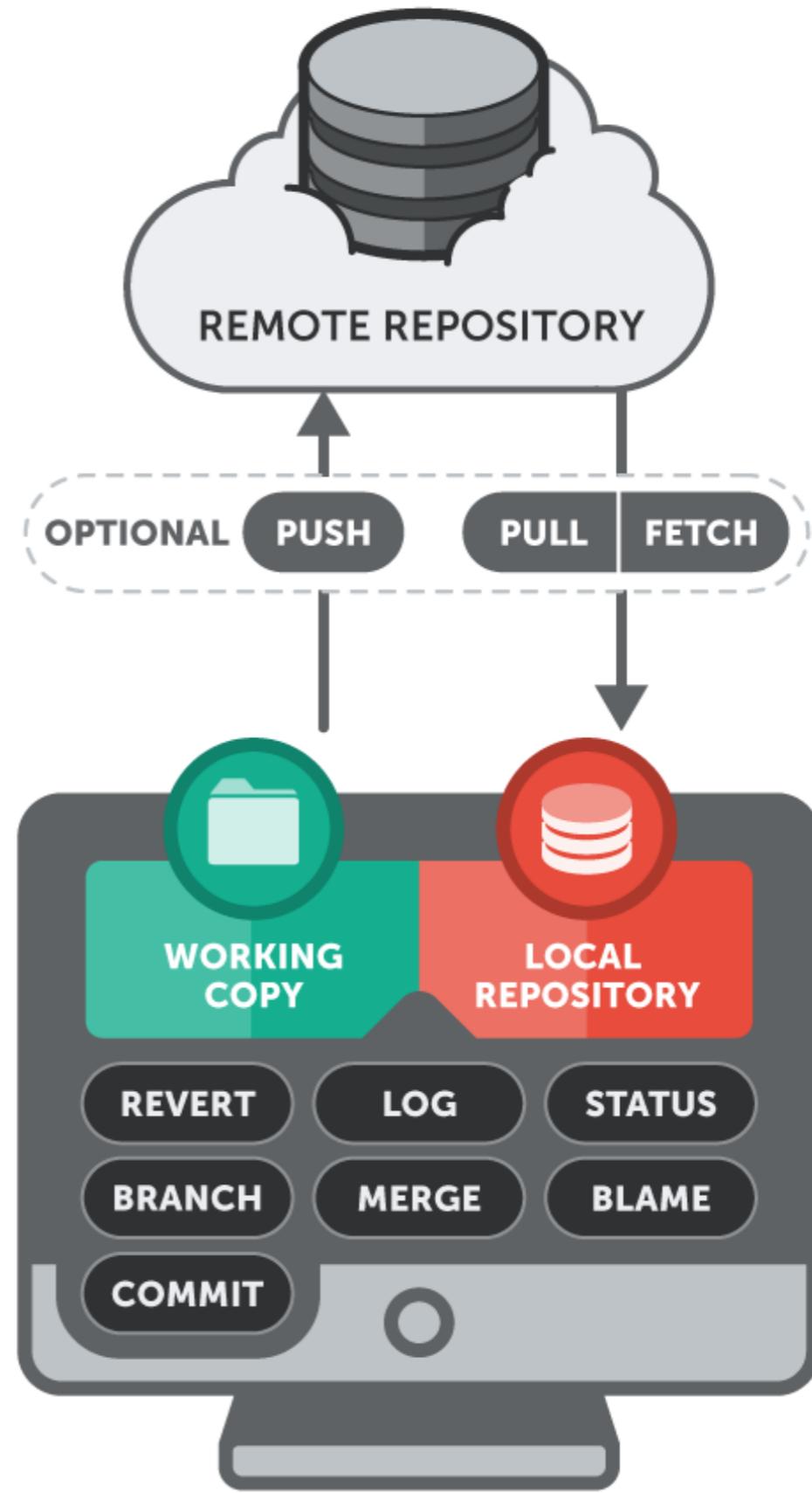
Criado por Linus Torvalds em 2005



SUBVERSION



GIT



Seguro

Rápido

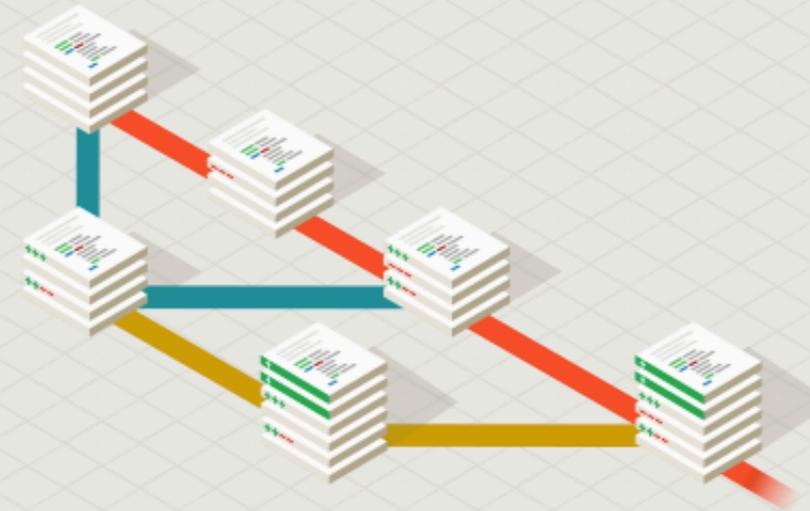
Colaborativo

O repositório da **Mozilla** que ocupava 12 GB utilizando o SVN e 3 GB usando CVS, foi reduzido para apenas 300 MB no Git

git --everything-is-local

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.



Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

[Mac GUIs](#)

[Tarballs](#)

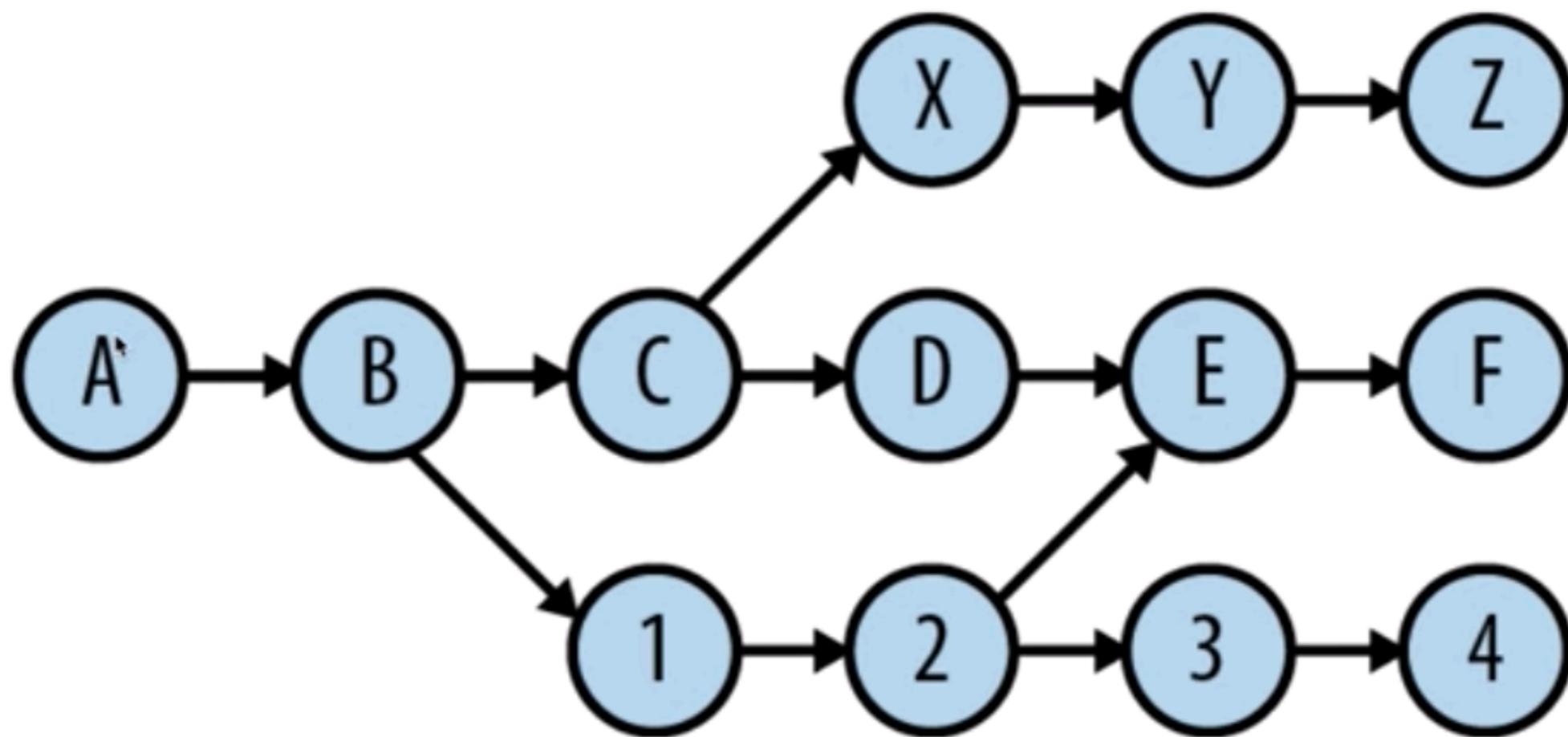


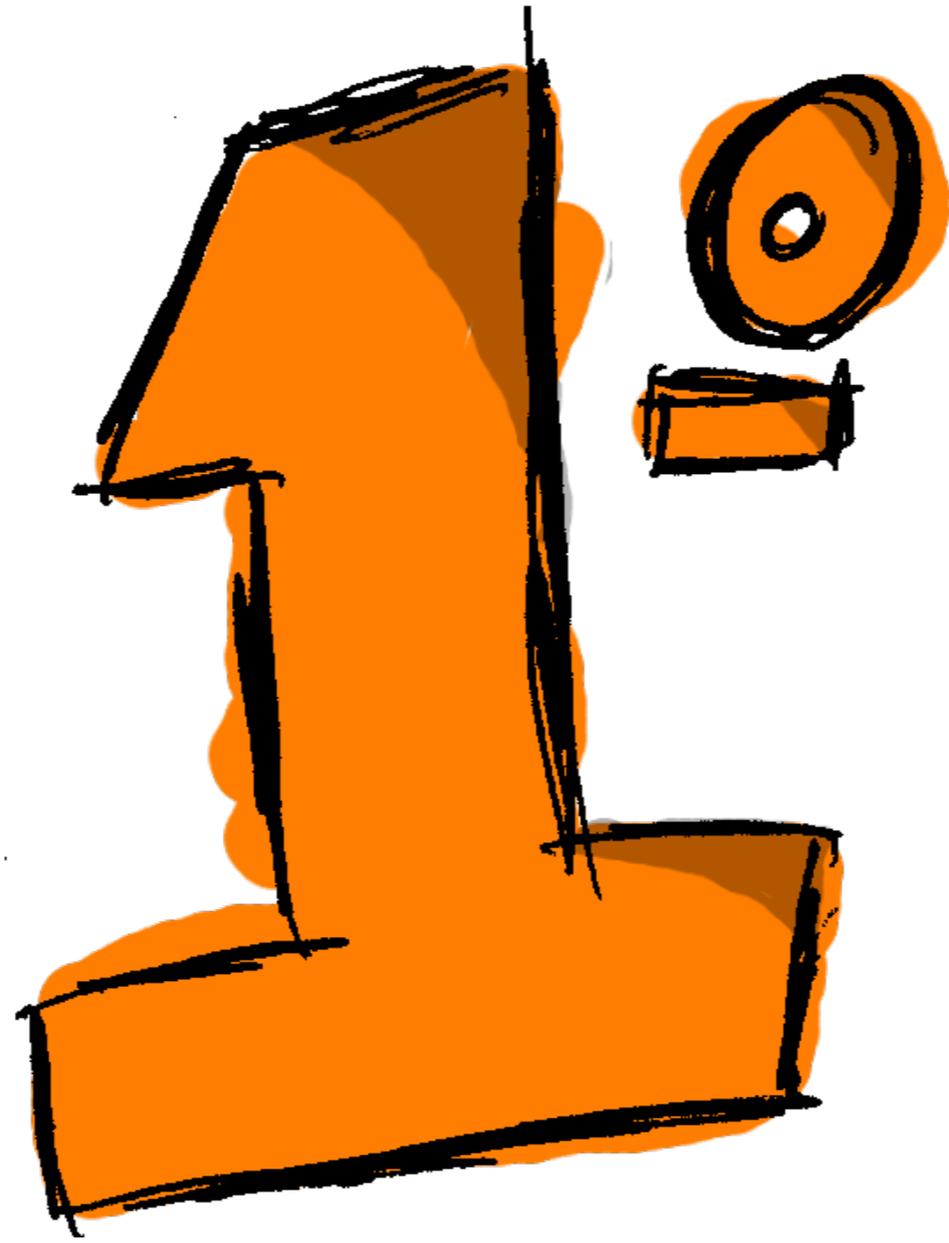
git init



**Armazena conteúdo,
não arquivos**

Representar toda a evolução
do seu código por meio de
um **grafo** de **commits**





git commit

working directory

git add

staging area

git commit

repository





working directory

git add

staging area

git commit

repository

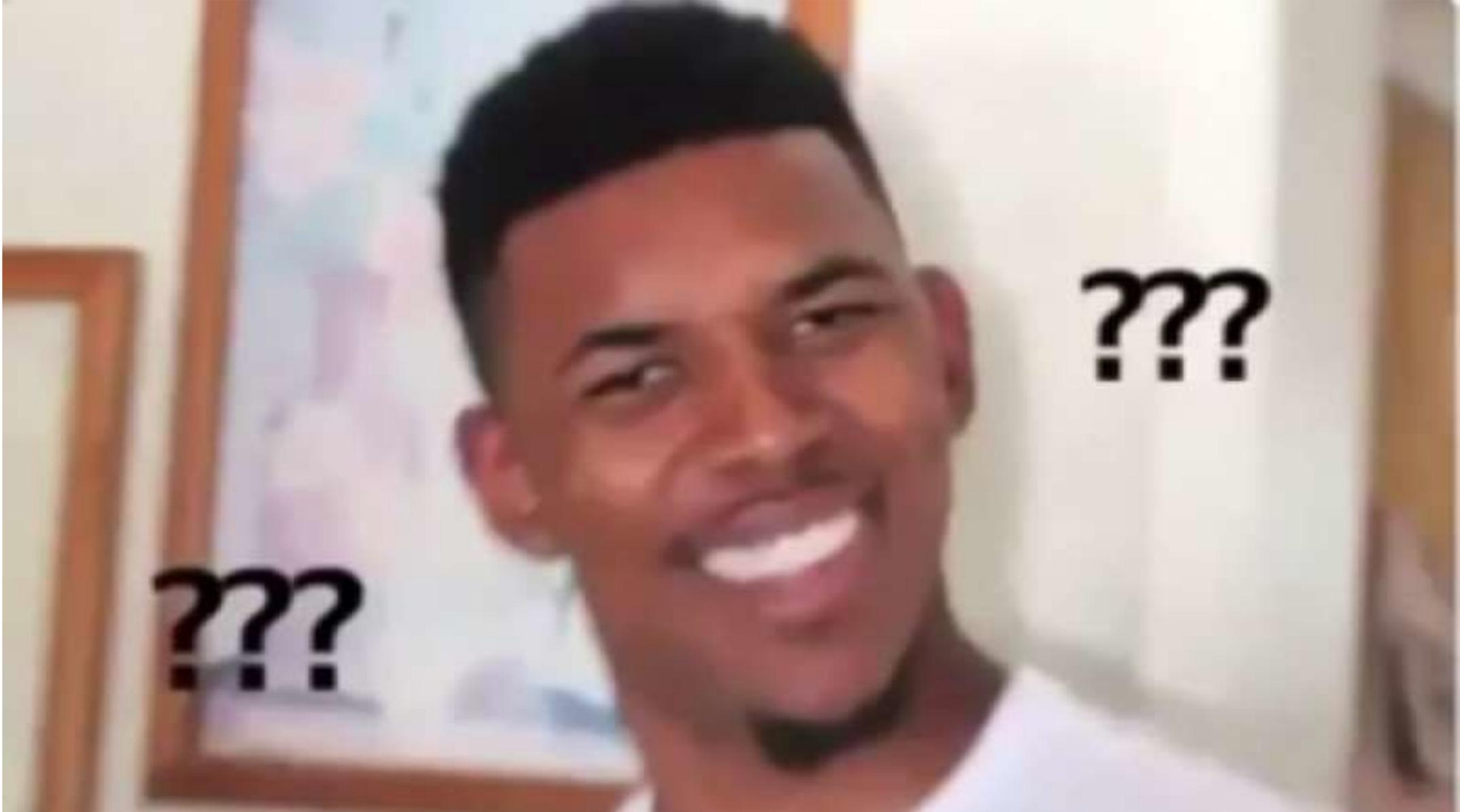


git status

git add

```
git add nome_arquivo
```

```
git add .  
git add -A  
git add --all
```



blob HASH
content

blob HASH
content

```
git commit -m "Primeiro commit"
```

commit HASH
tree HASH
parent

blob HASH
content

blob HASH
content

commit HASH
tree HASH
parent



tree HASH
HASH blob HASH arquivo
HASH blob HASH arquivo

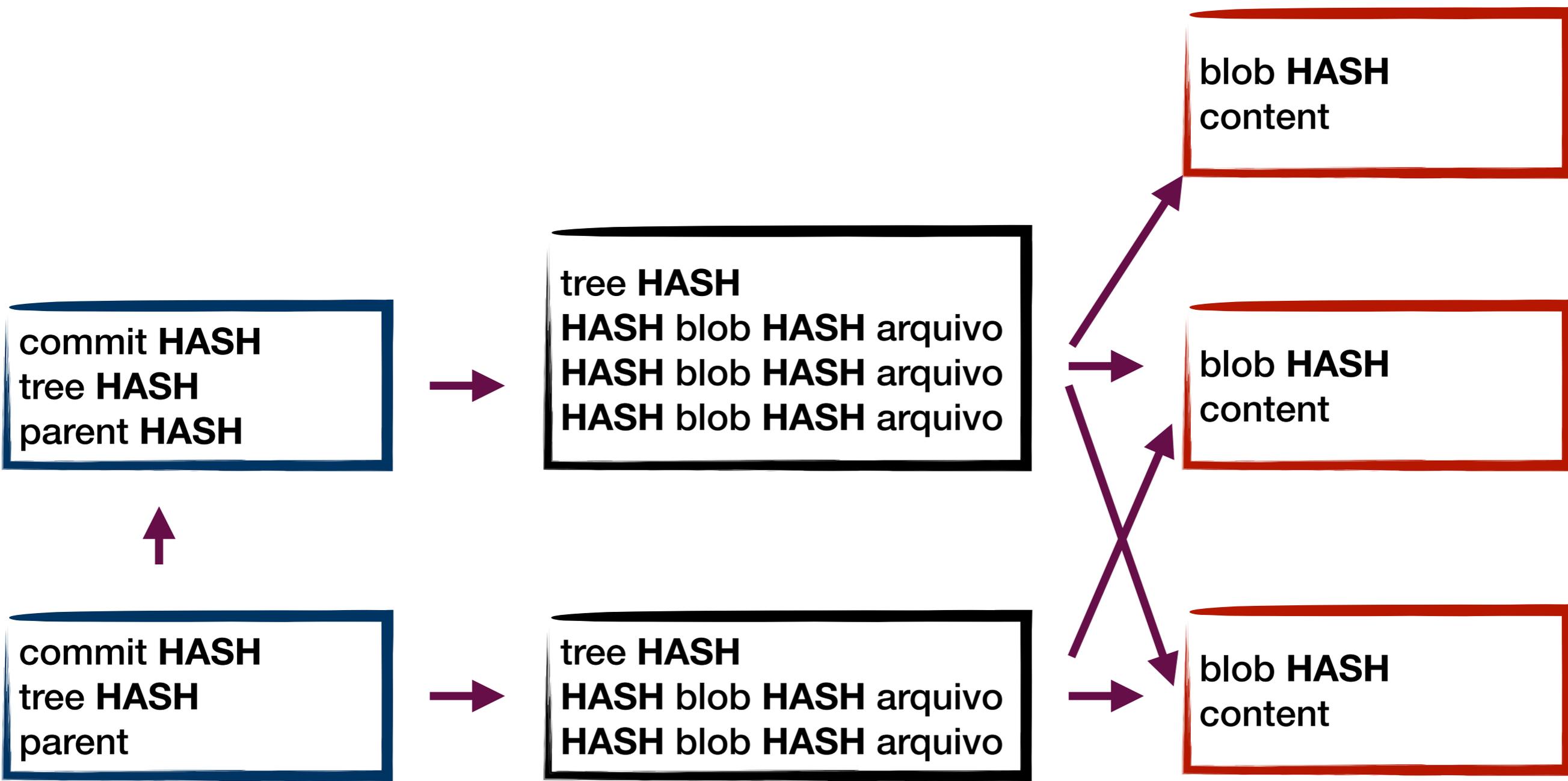


blob HASH
content



blob HASH
content

git log



version

init

status

add

commit

log