



git



GitHub



Mariana Gervino

Consultora de Qualidade de software  
Analista de Automação de Testes na B2W

Linkedin: <https://www.linkedin.com/in/mariana-gervino-b07582120/>

Github: <https://github.com/magervino>



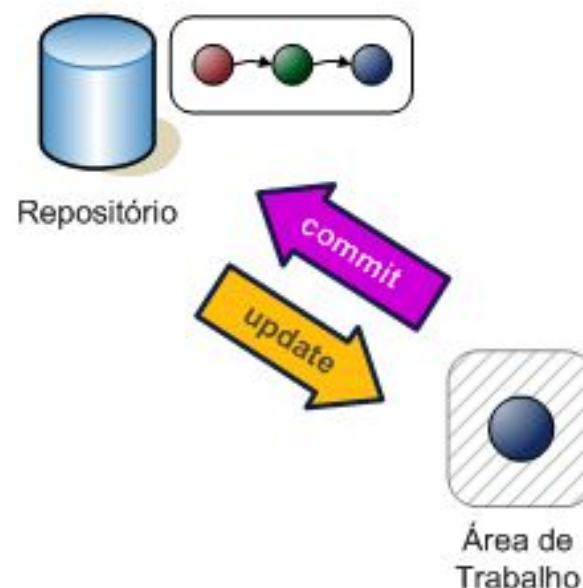
## O que veremos hoje

- O que é controlador de versão
- Por que usar?
- Git
- Github
- Link local com repositório remoto
- Comandos git hands on
- Resolvendo conflitos



## O que são Sistemas de Controle de Versão (VCS)

- Um sistema que consegue gerenciar as modificações feitas em um arquivo de forma que você possa recuperar versões específicas, possibilitando trabalho paralelo e concorrente de vários desenvolvedores sobre os mesmos arquivos e evitando que um sobrescreva o código do outro.
- Pode ser usado para registrar mudanças feitas em qualquer arquivo, não apenas códigos-fonte (exemplo: trabalho em grupo)





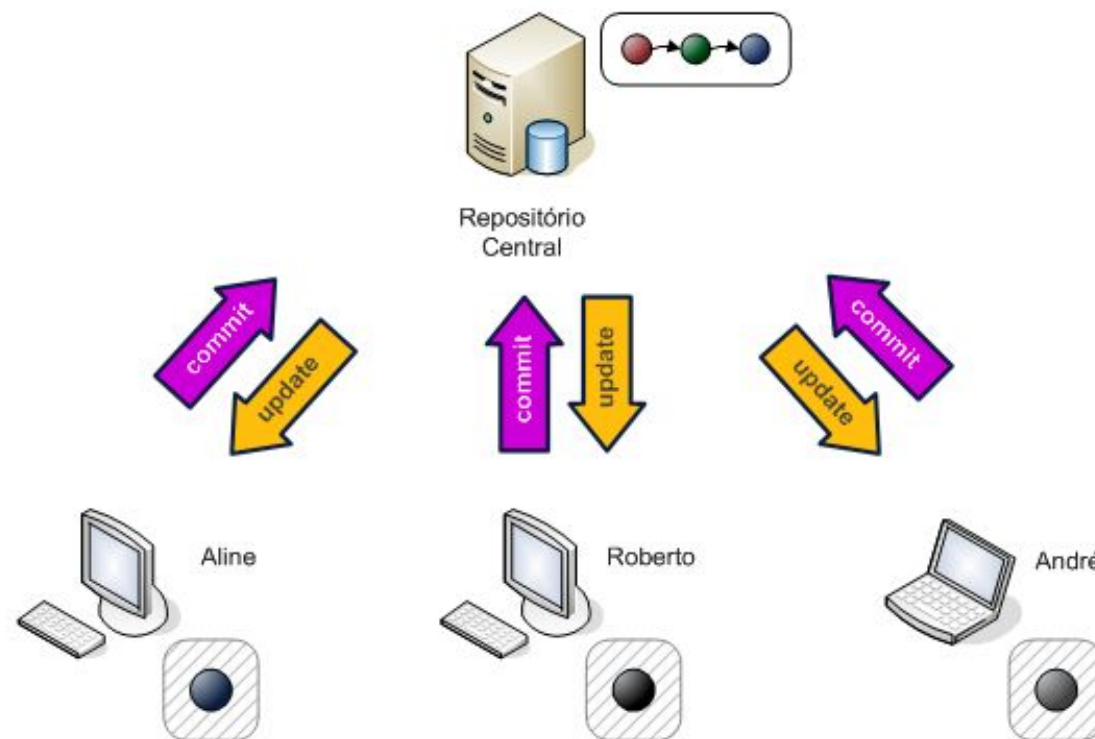
## **Por que usar Sistemas de Controle de Versão (VCS)?**

- Controle de histórico
- Trabalho em equipe
- Marcação e resgate de versões estáveis
- Rastreabilidade
- Organização
- Confiança



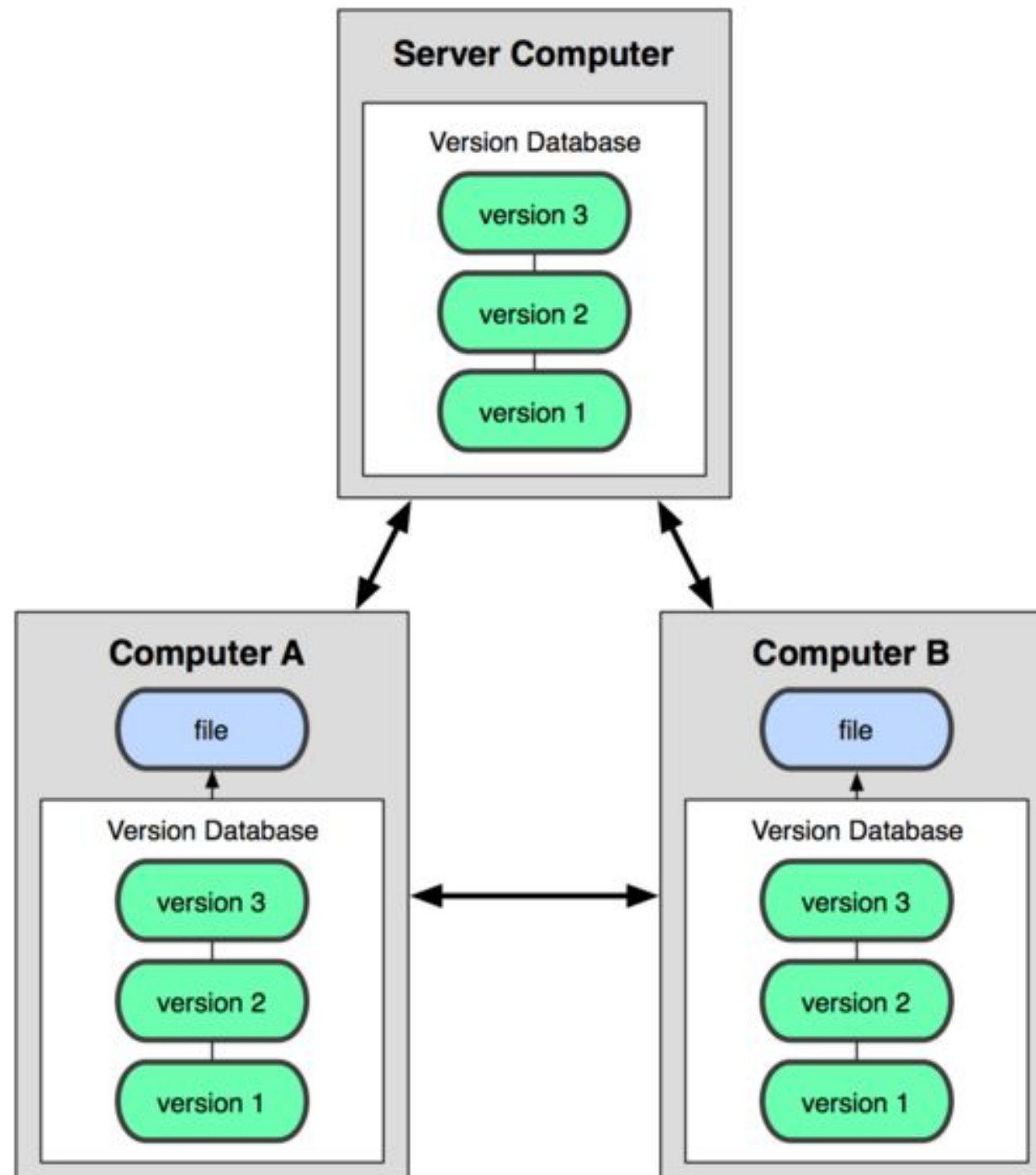
# Sistemas de Controle de Versão (VCS's) Centralizados - Exemplo: Subversion

Arquitetura Cliente - Servidor. Cópias de trabalho em cada máquina.





## Sistemas de Controle de Versão (VCS's) Distribuídos. Exemplo: Git





- Criado em 2005 pelo Linus Torvalds para a manutenção do Kernel do Linux.







insights.stackoverflow.com/survey/2018/#development-practices



Overview

Developer Profile

Technology

**Work**

I. Employment

II. Company Type

III. Career Values

IV. Ethics

V. Looking for a Job

VI. Job Priorities

**VII. Development Practices**

VIII. Salary

Community

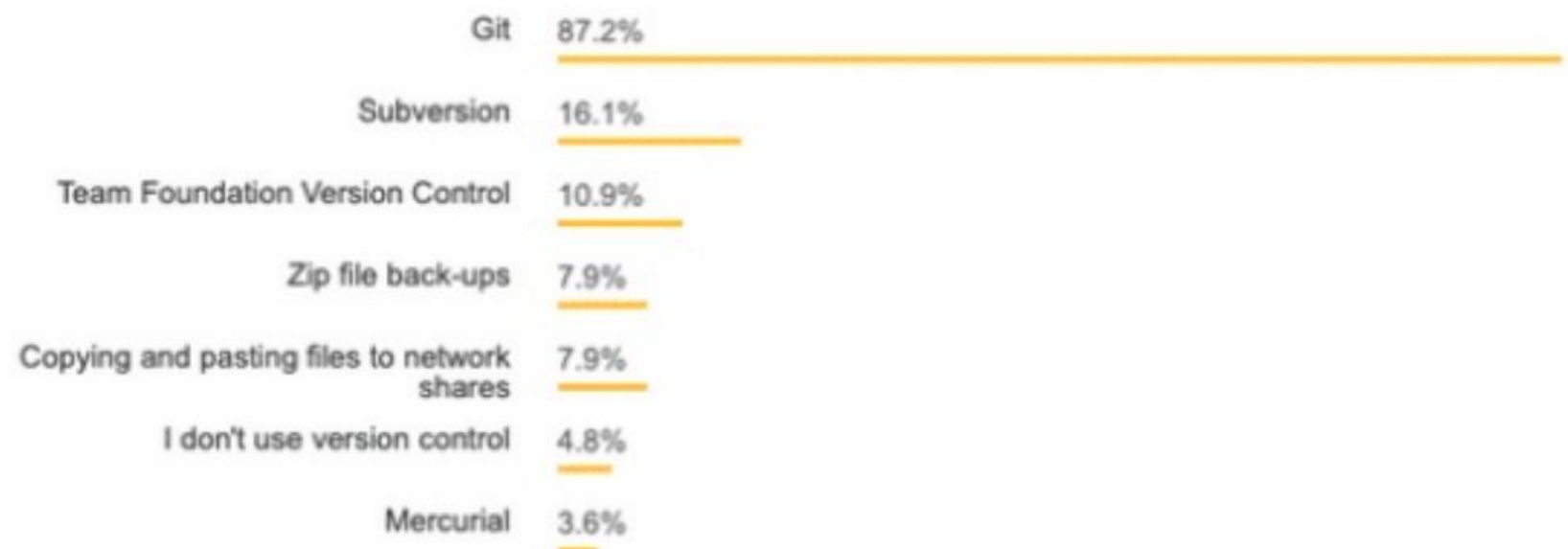
Methodology

Back to top

## Version Control

All Respondents

Professional Developers



74,298 responses; select all that apply

Git is the dominant choice for version control for developers today, with almost 90% of developers checking in their code via Git.



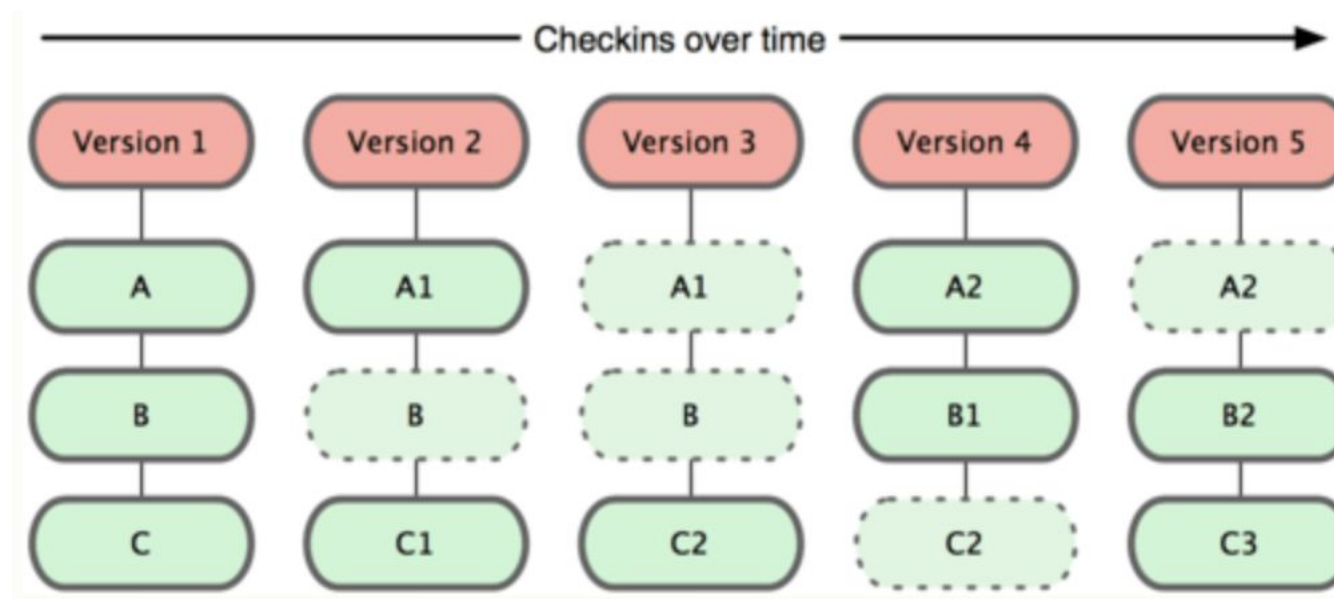
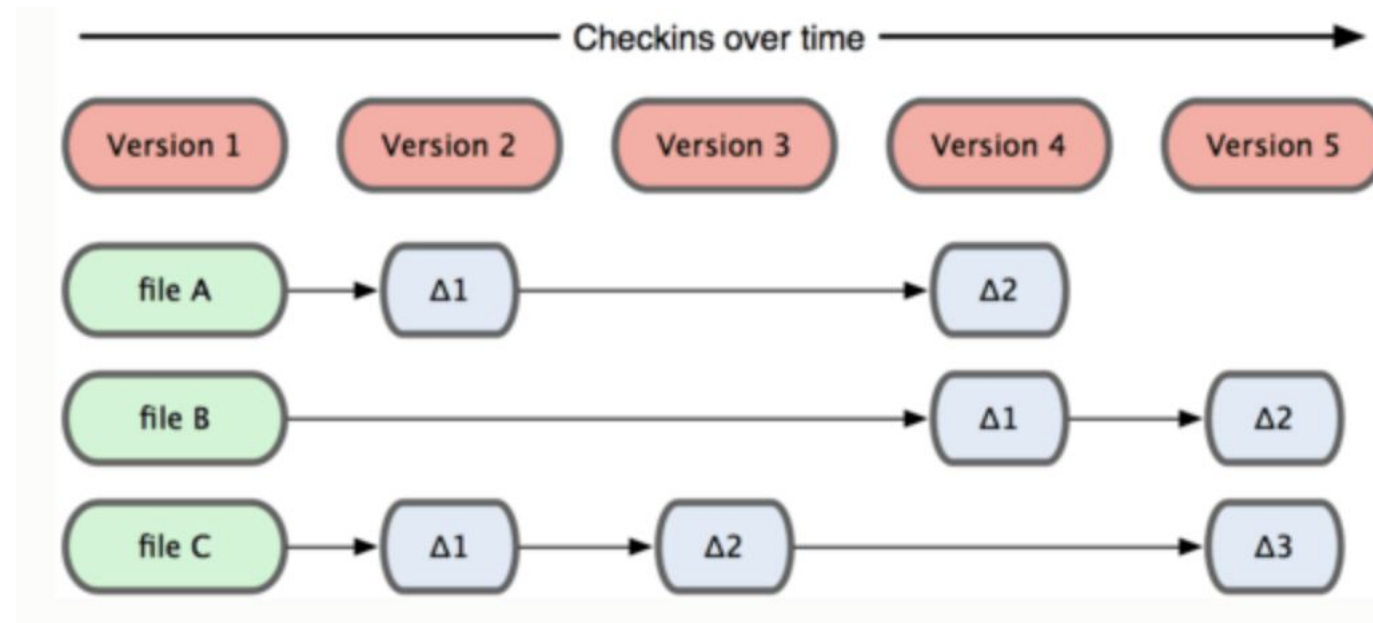
# Como funciona o Git

- instalando: [git-scm.com](https://git-scm.com)
- baseado em sistemas de arquivo (ou seja, tudo é pasta)





# Snapshots, não diferenças





## Repositório remoto

Servidor remoto que hospeda seu projeto, centralizando-o.

GitHub - gerencia users, permissões e projetos open source.  
Te permite adicionar colaboradores no repositório.

Hands on - Abrir o GitHub





## Comunicação entre seu computador e o repositório remoto - Github

Login ou chave SSH: chave criptografia pública.

- são geradas duas chaves: uma pública e uma privada.
- `ssh-keygen -t rsa` (instalar openssh-client para gerar a chave)

Hands on gerando a chave





# Entra na pasta

- *git init* - inicializa o repositório em um diretório;

```
git config --global user.name "SeuNome Sobrenome"  
git config --global user.email "seuEmail@gmail.com"
```



## Alguns comandos Git - **Hands ON**

**git init**: inicializar o git na pasta;

**git status**: checar status da pasta;

**git ls-files**: para ver os arquivos que o git está controlando

**git add**: adiciona os arquivos do working directory para a staging area ou index

**git commit -m**: adiciona os arquivos do index para o HEAD ou git directory com uma mensagem de commit

**git push**: envia o que está no HEAD para o repositório remoto

**git pull**: puxa o que está no repositório remoto para o local

**git clone**: clona todo o repositório remoto para local

**git remote**: lista os links remotos adicionados

**git remote add nomeDoRemote**: adiciona um repositório remoto

**git checkout -b**: atalho que cria uma branch nova e já vai para ela

**git checkout arquivo**: desfaz atualizações no arquivo para a última commitada no remoto

**git reset hashDoCommit**: volta todas modificações até um commit específico

**git revert hashDoCommit**: reverte o commit específico





# Resolvendo conflitos

```
aulaGit:~/Documents/git/magervino /propostas_homepage $ git push origin master
To https://github.com/magervino /propostas_homepage.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/jcfonsecaagit/propostas_homepage.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
aulaGit:~/Documents/git/magervino /propostas_homepage $
```





## Resolvendo conflitos - merge automático

```
Merge branch 'master' of https://github.com/magervino /propostas_homepage

# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
```

```
aulaGit:~/Documents/git/magervino /propostas_homepage $ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From https://github.com/magervino /propostas_homepage
 * branch                master      -> FETCH_HEAD
   f4aa10e..b1d9c4a  master      -> origin/master
Auto-merging proposta_1.html
Merge made by the 'recursive' strategy:
 proposta_1.html | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```



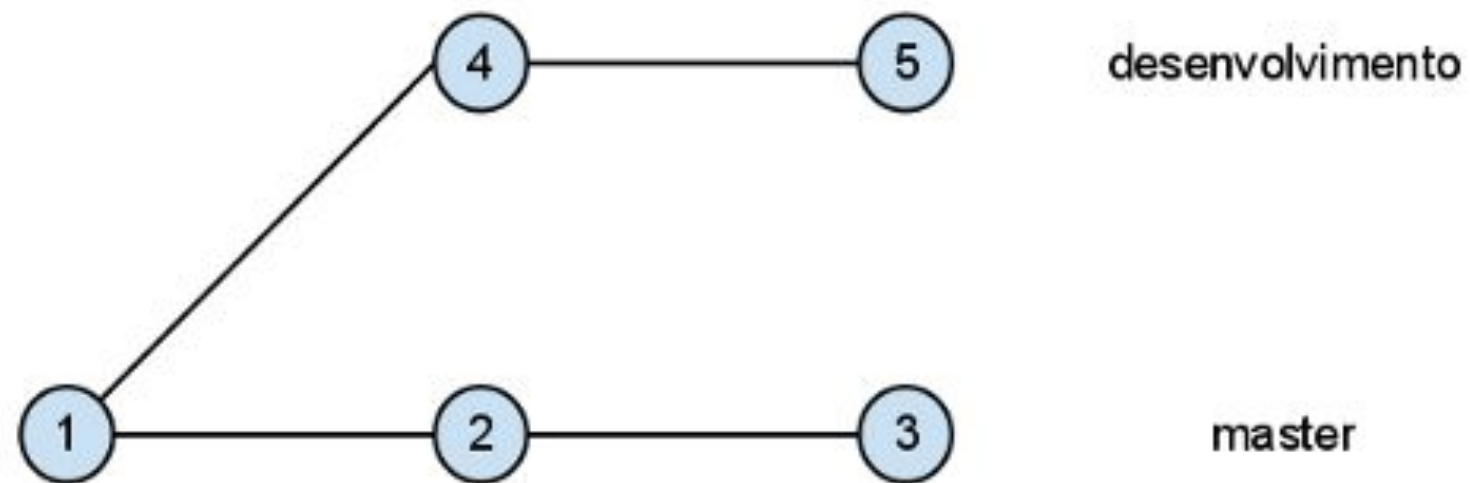
## Resolvendo conflitos - merge manual

```
aulaGit:~/Documents/git/magervino /propostas_homepage $ git pull
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 2), reused 5 (delta 1)
Unpacking objects: 100% (6/6), done.
From github.com:magervino /propostas_homepage
   b1d9c4a..548efa5 master -> origin/master
Auto-merging proposta_1.html
CONFLICT (content): Merge conflict in proposta_1.html
Automatic merge failed; fix conflicts and then commit the result.
Alura:~/Documents/git/magervino /propostas_homepage $
```

```
1  <html>
2    <head>
3      <title>Proposta um para site da empresa</title>
4    </head>
5    <body>
6      <h1>Cabeçalho do sistema</h1>
7
8      <main>
9        <<<<<<< HEAD
10          Isso aqui é o conteúdo do site
11        </main>>
12      =====
13          Isso aqui é o conteúdo da página
14        </main>
15      >>>>>>> 548efa50f51d90fc2727bc67fcb09184d0811539
16
17      <h1>Rodapé do sistema</h1>
18    </body>
19  </html>
20
```

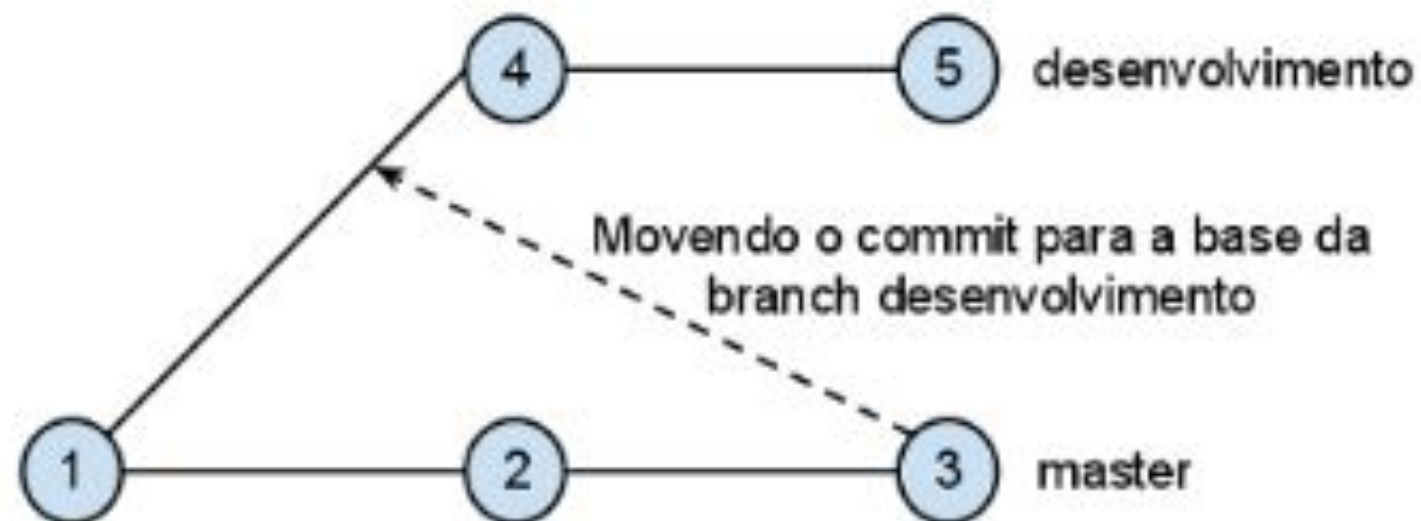


# Evitando conflitos



## git rebase master

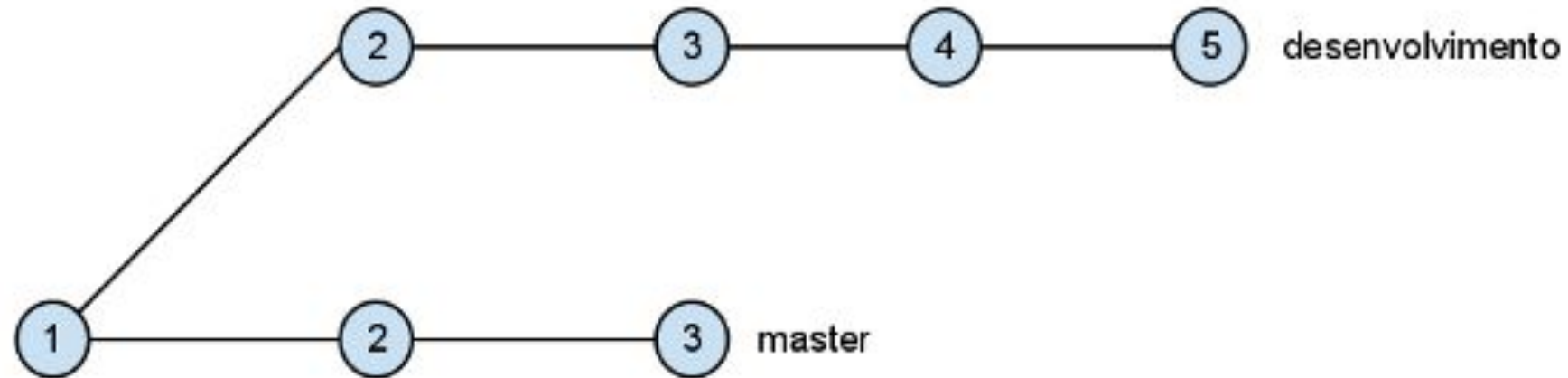
de dentro da branch desenvolvimento



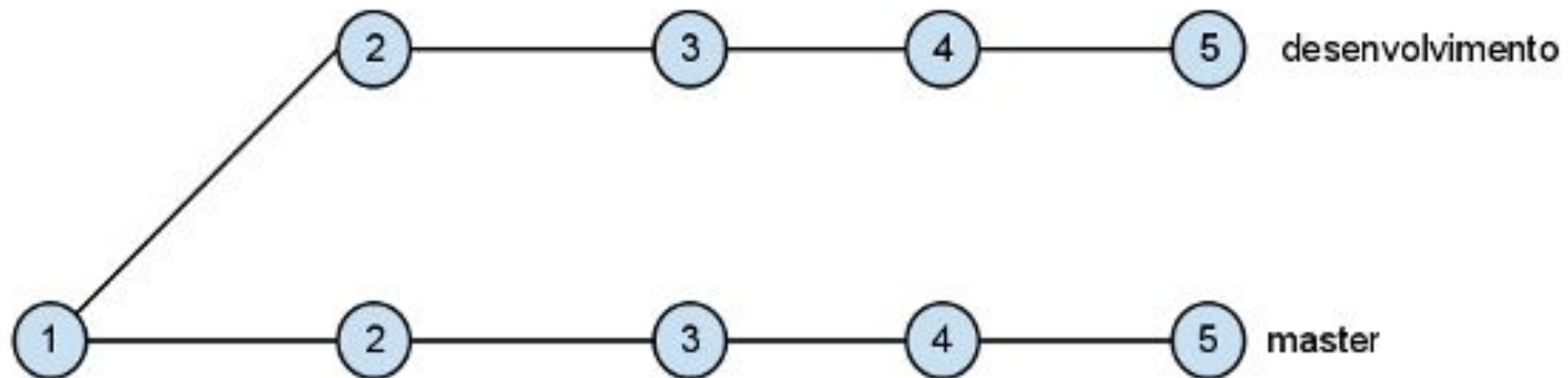




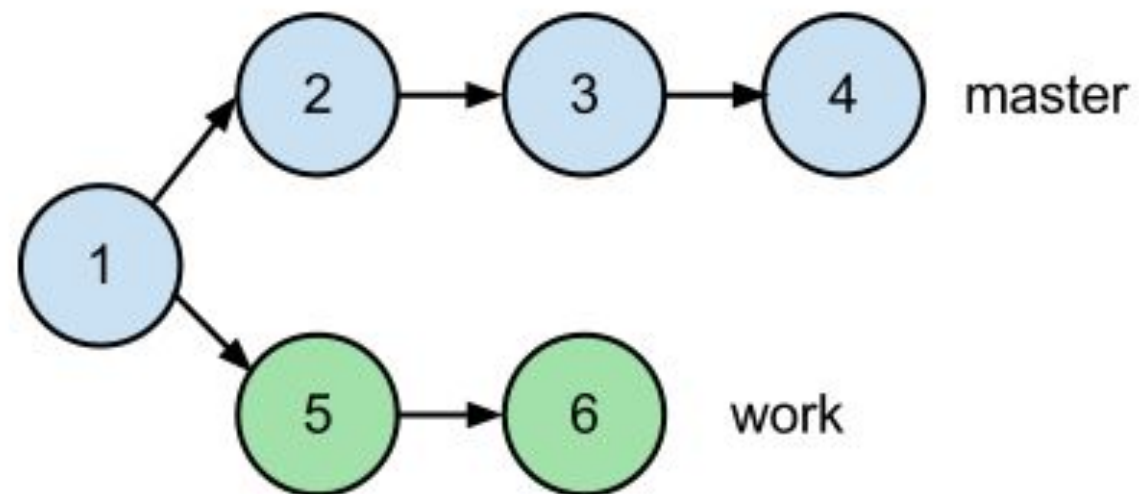
# Evitando conflitos



**git merge desenvolvimento**  
de dentro da master

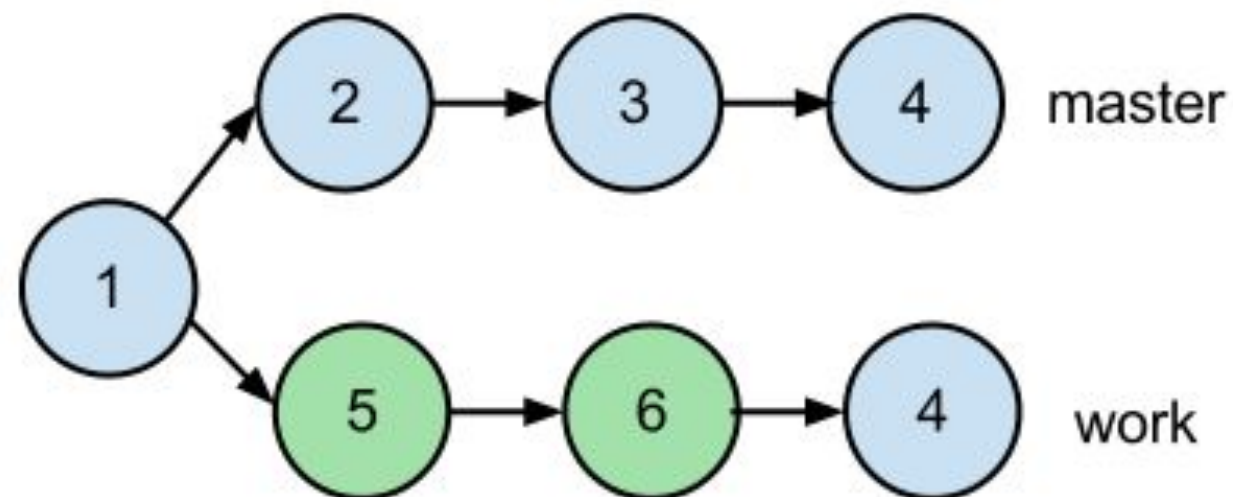


# Cherry-pick



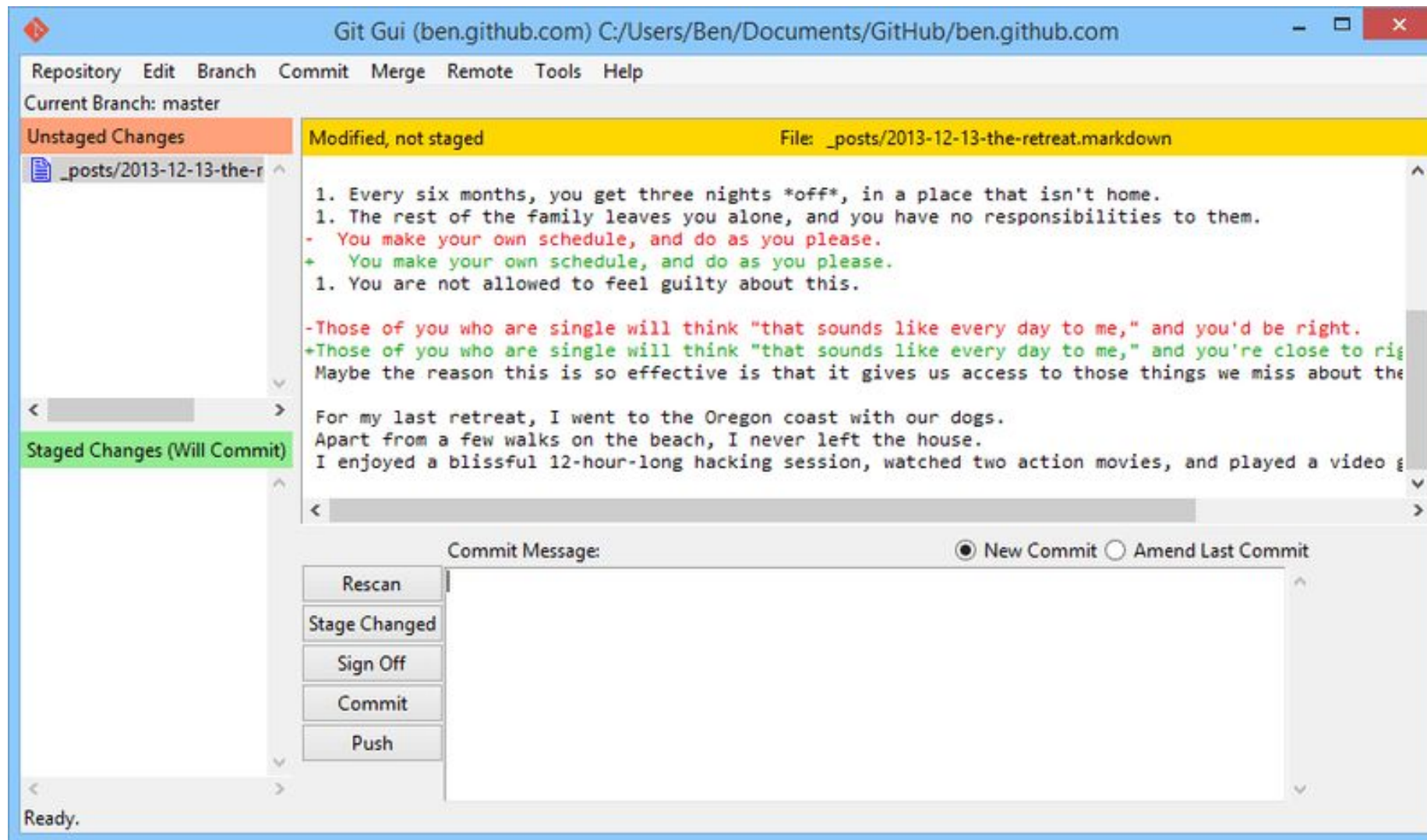
## git cherry-pick

nesse caso, de dentro da branch work





# Interface Gráfica para Git - Git Gui



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



**Obrigada! =)**