
GIT

Controle de Versão

- VCS (Version Control System)

Para que serve?

- Gerenciamento de Projetos em Equipe
 - Resgate de Versões Antigas
 - Controle do Histórico de Modificações
-

Tipos de Controle de Versão

VCS Centralizado

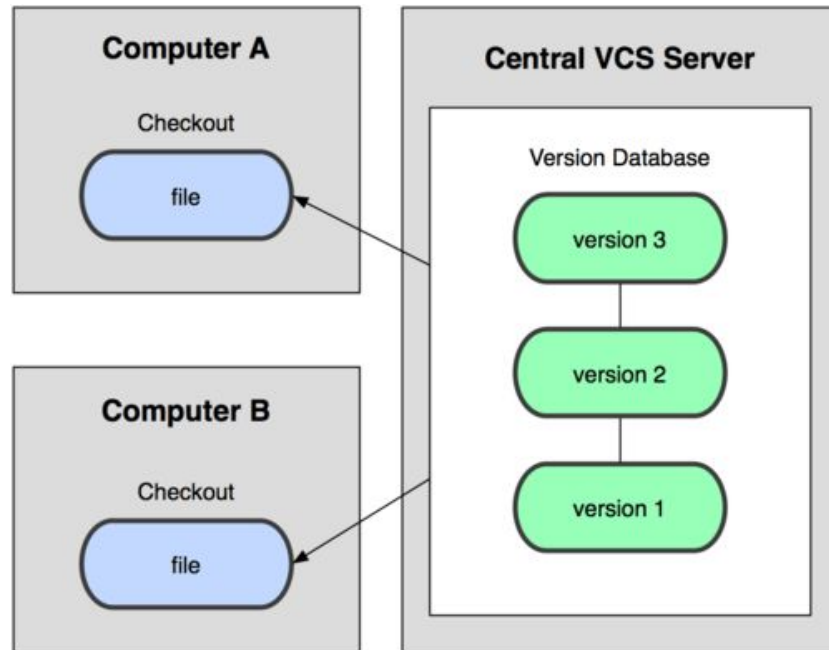
Vantagens:

-Intuitivo

Desvantagens:

-Utiliza o arquivo original

-Conflito de versões



Tipos de Controle de Versão

VCS Distribuído

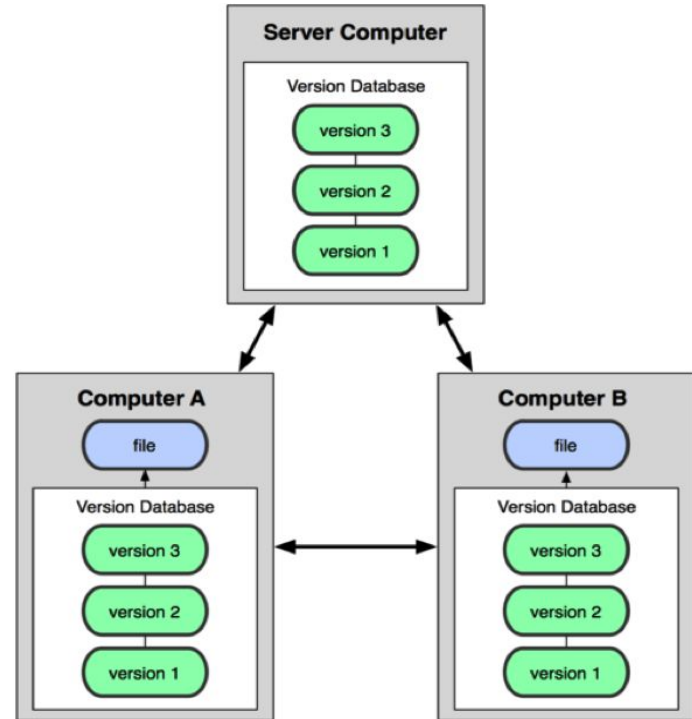
Vantagens:

-Utiliza cópias dos arquivos

- Permite “Voltar no tempo”

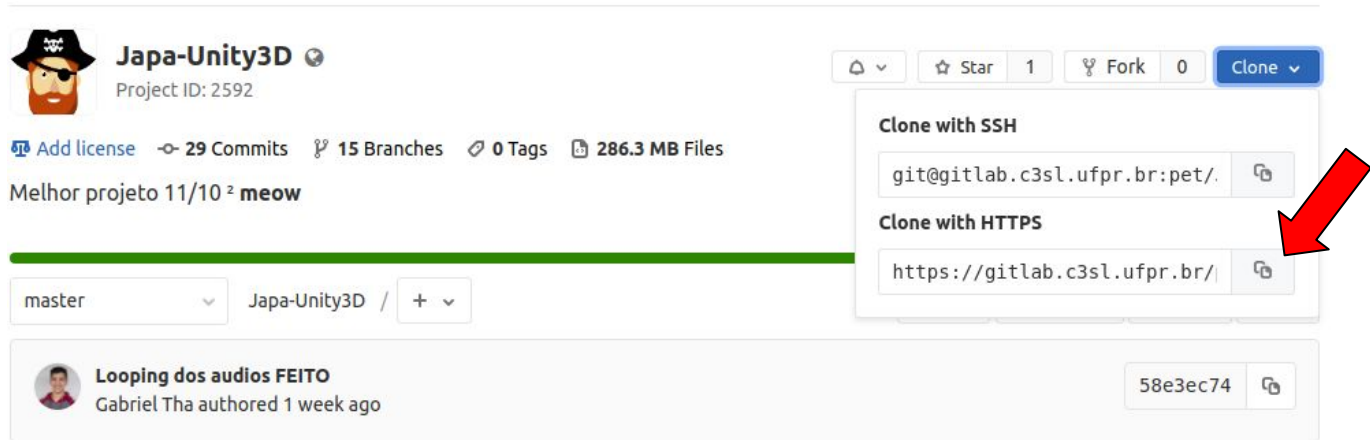
Desvantagens:

-NENHUMA



GIT Clone

- Faz uma cópia do Repositório (projeto) para o seu computador



The screenshot shows the GitLab interface for a repository named 'Japa-Unity3D'. The repository owner's profile picture is a cartoon character with a red beard and a black hat. The repository details include 'Project ID: 2592', 'Add license', '29 Commits', '15 Branches', '0 Tags', and '286.3 MB Files'. A green progress bar is visible. Below the repository name, there is a dropdown menu showing 'master' and 'Japa-Unity3D / +'. A red arrow points to the 'Clone with HTTPS' option in the dropdown menu, which shows the URL 'https://gitlab.c3sl.ufpr.br/'.

Japa-Unity3D Project ID: 2592

[Add license](#) [29 Commits](#) [15 Branches](#) [0 Tags](#) [286.3 MB Files](#)

Melhor projeto 11/10² **meow**

master Japa-Unity3D / +

Clone with SSH
git@gitlab.c3sl.ufpr.br:pet/

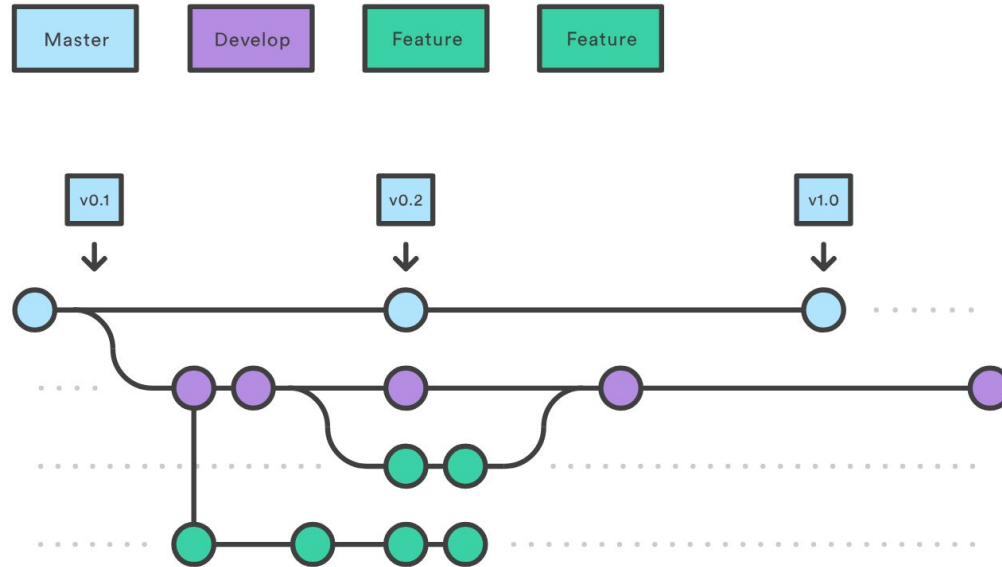
Clone with HTTPS
https://gitlab.c3sl.ufpr.br/

Looping dos audios FEITO
Gabriel Tha authored 1 week ago 58e3ec74

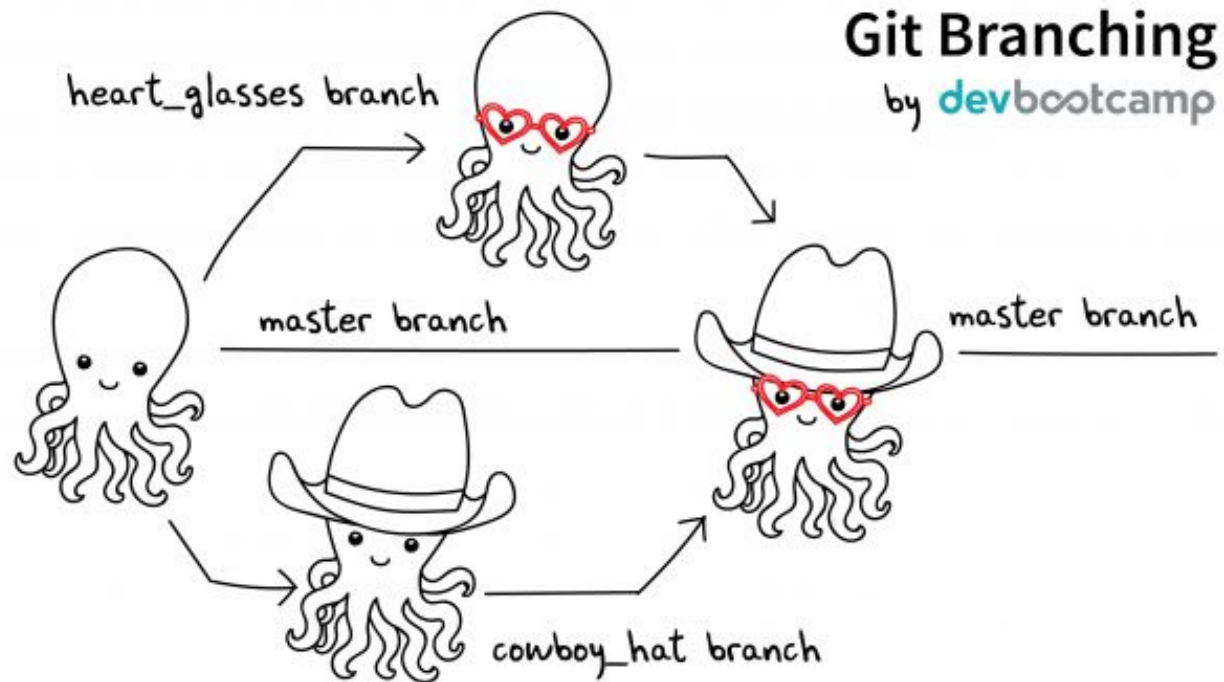
- `git clone <endereço HTTPS>`
-

GIT Branch

As Branches são ramificações do projeto principal



GIT Branch



GIT Branch

Como navegar entre Branches:

1. `git checkout <nome de destino>`

Como criar uma nova Branch:

1. Estar dentro do Diretório do Repositório
 2. `git branch <nome da branch nova>`
 3. `git checkout <nome da branch nova>`
 4. `git push origin <nome da branch nova>` (envia p/ remoto)
 5. `git push origin :<nome da branch>` (remove remota)
-

GIT Push & Pull

- **PUSH**

Envia suas alterações para o repositório remoto

```
git push (-u origin <branch>)
```

- **PULL**

Atualiza seus arquivos locais de acordo com o que está no repositório remoto

```
git pull
```

GIT Status

Mostra sua branch atual 

Mostra os arquivos modificados e não adicionados/commitados 

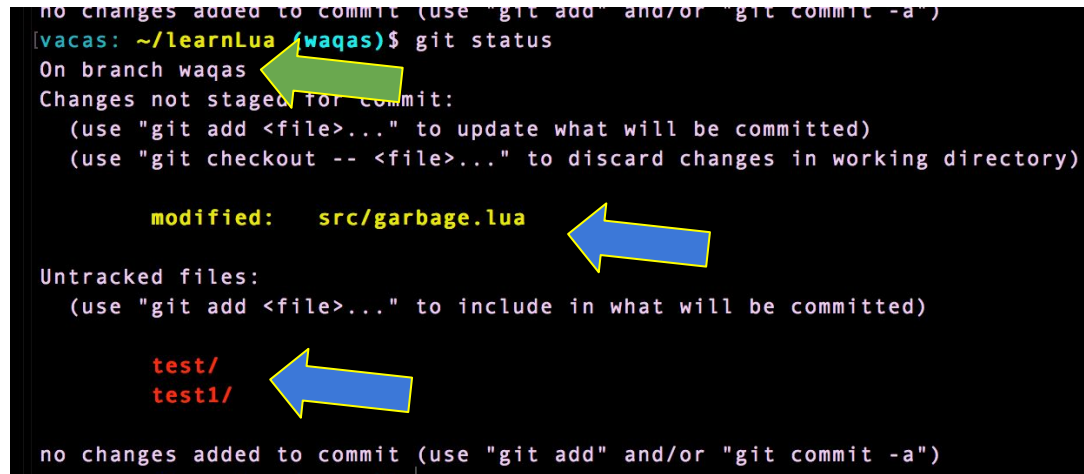
```
no changes added to commit (use "git add" and/or "git commit -a")
[vacas: ~/learnLua (waqas)]$ git status
On branch waqas
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/garbage.lua

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test/
        test1/

no changes added to commit (use "git add" and/or "git commit -a")
```



GIT Add

- Adiciona suas modificações de arquivo

Como adicionar modificações:

1. `git add <nome do arquivo>` *ou* `git add .`

GIT Commit

-Descreve suas modificações no repositório

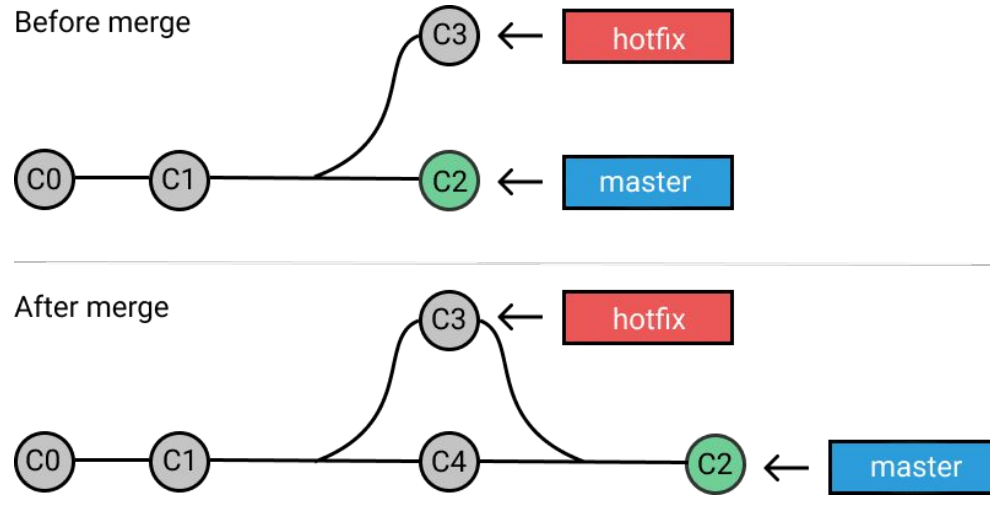
Como comentar modificações:

1. `git commit -m "<descrição>"`
-

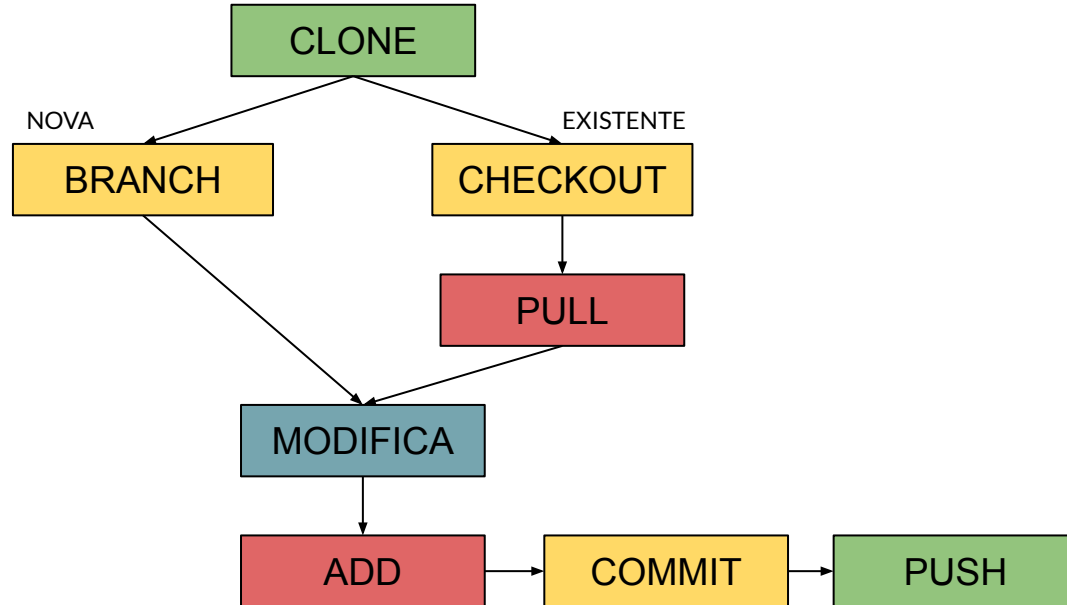
GIT Merge

-Faz a mesclagem de duas Branches

- git merge <nome da branch que você quer juntar com a atual>



PASSO A PASSO DE UTILIZAÇÃO



Tips

```
git config --global/local user.name "seu_user"
```

```
git config --global/local user.email "seu_email"
```

```
git log --graph --all
```

GIT WORKFLOW: <https://br.atlassian.com/git/tutorials/comparing-workflows>

COMMIT GUIDE: <https://gist.github.com/robertpainsi/b632364184e70900af4ab688decf6f53>
