



GIT E GITHUB

Gabrielle Amorim Porto

GIT

- Sistema de controle de versão de arquivos;
- Facilita o desenvolvimento colaborativo de software;
- Permite que um mesmo arquivo seja editado ao mesmo tempo por pessoas diferentes, sem sobrescrevê-lo;
- Maior facilidade ao fazer deploy.



GIT

- Registra alterações em arquivos ao longo do tempo;
- Permite voltar a uma versão antiga dos arquivos;
- Registra quem fez alterações nos arquivos.



GITHUB

- Rede social para desenvolvedores;
- Permite que usuários hospedem seus projetos Git nas nuvens;
- Fornece funcionalidades extras aplicadas ao Git – fork, pull request;
- Open-source.

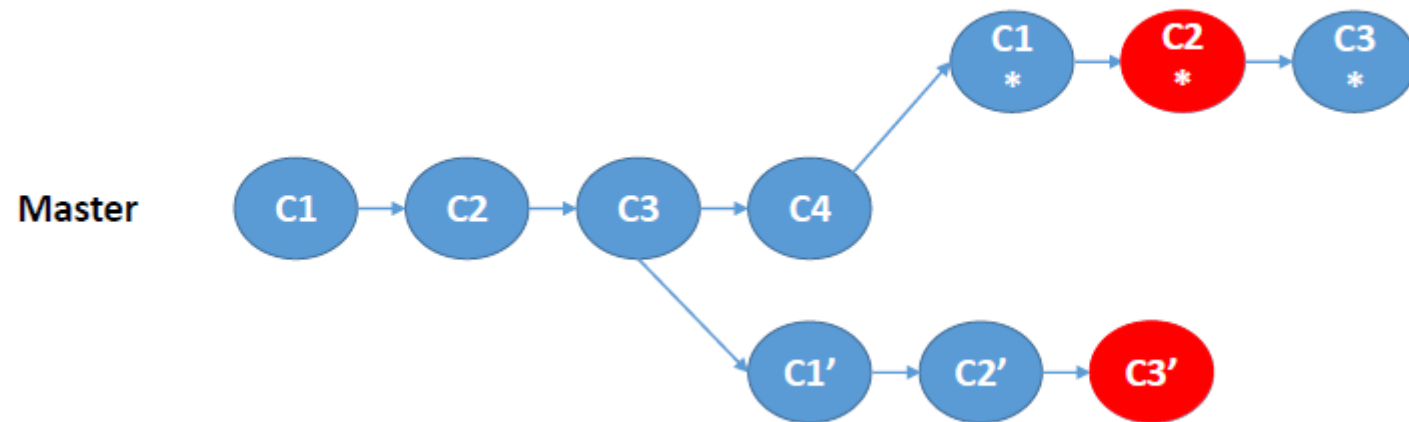


CONCEITOS GITHUB

- **Fork** – ao clicar em fork no repositório de alguém, ele pega o repositório dessa pessoa e copia para a sua conta. Lá, você pode editar esses arquivos e depois devolver a essa pessoa com as suas edições.
- **Pull request** – envia suas contribuições para o repositório principal, onde vai ficar pendente até a aceitação.

CONCEITOS GIT

- **Commit:** registra um conjunto de alterações feitas pelo desenvolvedor, cada commit possui uma hash única associada ao mesmo.
- **Branch:** é uma linha independente de desenvolvimento criada a partir do branch principal, o master.



ESTÁGIOS DO GIT

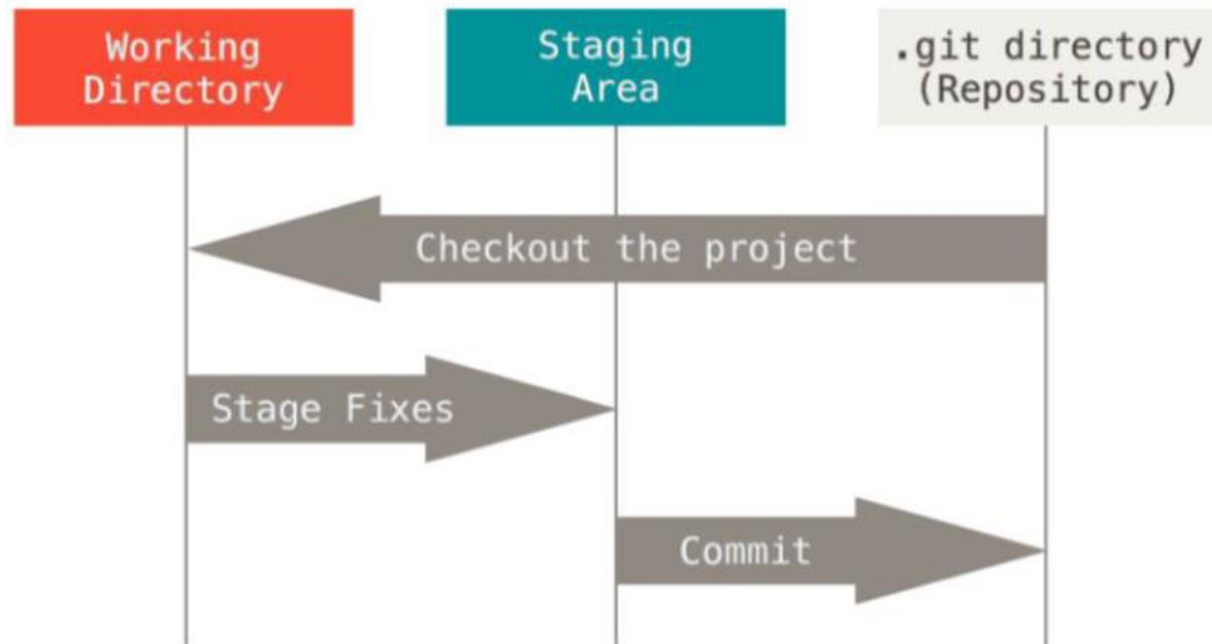
- **Modified:** arquivos alterados, mas ainda não foram commitados;
- **Staged:** arquivos modificados que foram assinalados para o próximo commit que você fizer;
- **Committed:** dados armazenados corretamente no repositório local.

CONCEITOS GIT

- **Conflito de merge:** duas pessoas alteraram a mesma linha do mesmo arquivo, qual alteração prevalece?
- Ocorre quando o repositório local está desatualizado;
- Necessita de intervenção manual do desenvolvedor.

```
<<<<<<< HEAD: nomeArquivo
if (a > b) { ... }
=====
if (b > a) { ... }
>>>>>>> hash do commit
```


CONCEITOS GIT



- Altera os arquivos no **working directory**;
- **Staging area** é o arquivo que armazena informações sobre o que vai ser incluído no próximo commit;
- **.git directory** registra as alterações permanentemente após o commit.

COMANDOS GIT

- **git clone url** – copia todos os arquivos de um repositório para a máquina local;
- **git init** – inicializa um repositório git;
- **git status** – apresenta os arquivos que foram modificados;
- **git add nomeArquivo** – adiciona o arquivo ao staging area;

COMANDOS GIT

- **git commit -m “mensagem do commit”** – faz o commit dos arquivos que foram colocados na staging area;
- **git log** – mostra o histórico de commits;
- **git reset HEAD nomeArquivo** – remove o arquivo da staging area;
- **git remote -v** – apresenta os “remotes” adicionados;
- **git remote add nomeRemote urlRemote** – adiciona um novo remote;

COMANDOS GIT

- **git fetch nomeRemote** – compara o seu remote (origin) com o remote passado;
- **git push nomeRemote nomeBranch** – envia as suas alterações para um remote;
- **git checkout -b nomeBranch** – cria um novo branch;
- **git checkout nomeBranch** – altera para um branch já criado;
- **git branch** – mostra todos os branches existentes;

COMANDOS GIT

- **git merge nomeBranch** – realiza o merge. Em alguns casos, também é possível realizar o merge no Github;
- **git pull nomeRemote nomeBranch** – atualiza o repositório local;

TRABALHANDO COM O GIT E GITHUB

- Instalar o git: <https://git-scm.com/download/win> ou `sudo-apt get install git`
- Criar conta no github: <https://github.com/join>
- Configurar o git:
 - `git config --global user.name "Gabrielle"`
 - `git config --glocal user.email gabrielleporto19@gmail.com`

TRABALHANDO COM O GIT E GITHUB

- Realizar o fork de um repositório e cloná-lo, ou inicializar um novo repositório git;
- Criar um novo branch;
- Realizar alterações necessárias;
- Adicionar os arquivos alterados na staging area;
- Realizar commit;
- Realizar push;
- Realizar pull request;
- os **commits** agrupam arquivos associados;
- um **pull request** pode possuir mais de um commit.