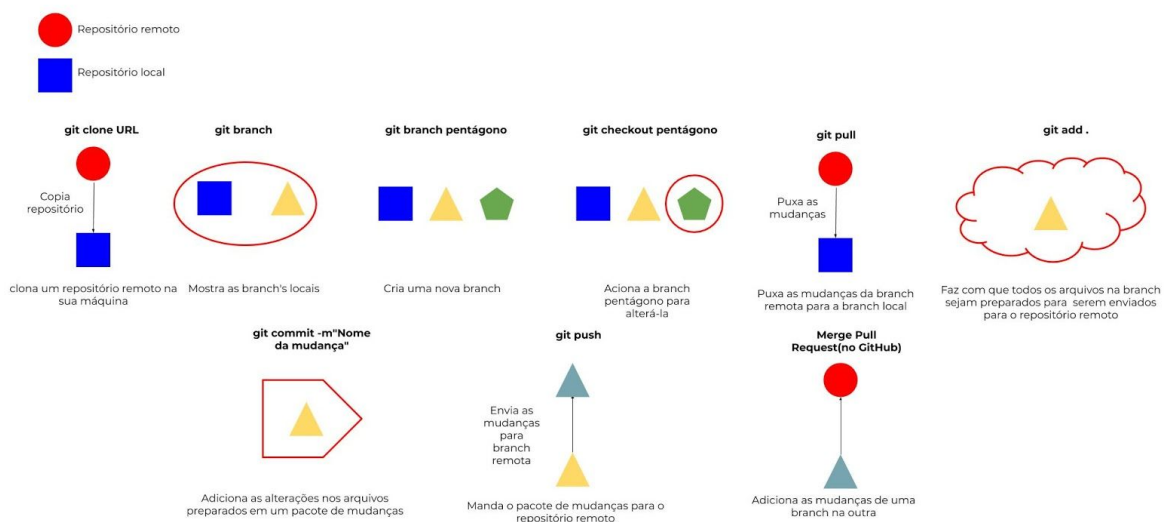


Introdução

Esse texto tem o objetivo de exemplificar como o Git e GitHub funcionam. É interessante ter uma visão geral disso e depois ir entrando em cada etapa do mesmo. Assim esse conteúdo vai ser cíclico. Então se não entenderem algum conteúdo logo de primeira a gente vai voltar nele após.

Roteiro 1



Um **repositório remoto** no GitHub é onde são guardados os seus arquivos na nuvem.

Já o **repositório local** é onde se guarda os seus arquivos na sua máquina.

Branch é uma versão de um código.

- **git clone** - Serve para "fazer o download" de um repositório remoto para a sua máquina.
- **git branch** - Se usa este comando para ver as branches, de um repositório, que existem na sua máquina.
- **git branch nome-da-nova-branch** - Serve para criar uma nova branch
- **git checkout nome-da-branch** - É usado para acessar (apontar para) uma branch específica. Essa branch pode ser tanto a nova, como alguma outra branch (não necessariamente local)
- **git pull** - Serve para puxar para a sua máquina mudanças feitas ao código salvas no repositório remoto, que no nosso caso está no GitHub.

- **git add . / git add path/to/file** - Utiliza-se esse comando para colocar as mudanças no index, que é como se fosse um aeroporto: As mudanças ainda não saíram para a nuvem, mas já estão prontas para isso.
- **git commit -m "mensagem"** - Serve para juntar as mudanças do índice num pacote de mudanças (chamado de commit) e inseri-las no repositório local.
- **git push** - Serve para passar as mudanças para o repositório remoto. Você "faz o upload" das mudanças da sua máquina para o repositório remoto.
- **Merge Pull Request** - Serve para agregar as mudanças feitas pela sua branch a uma outra branch. Quando se quer oficializar uma alteração no código, faz-se o merge da branch deste código com a branch master.



This branch has no conflicts with the base branch

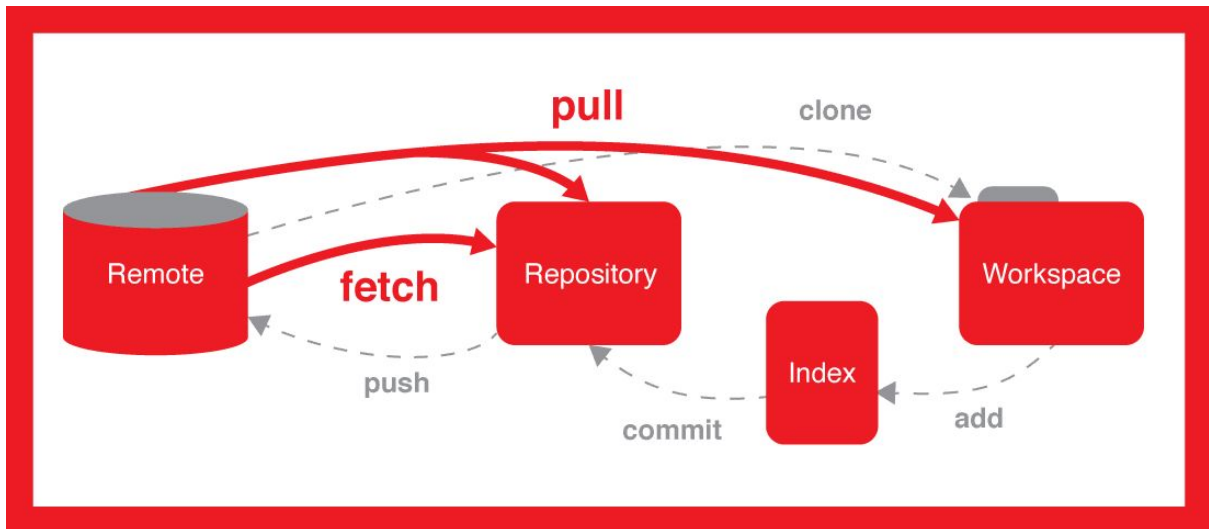
Merging can be performed automatically.

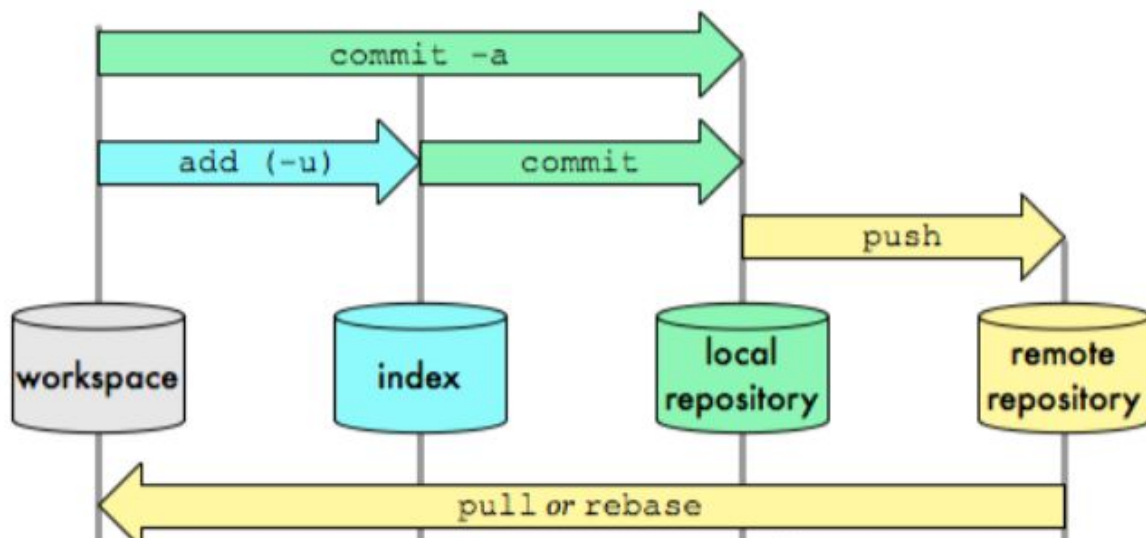
Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Roteiro 2

Tente entender melhor como funciona os comandos do GitHub a partir dessas imagens.





Roteiro 3

Primeiro, vamos procurar algum repositório para clonar. Você ainda se lembram o que é isso? Bem, explicando rapidamente o que é clonar um repositório: Clonar significa copiar um repositório remoto (na nuvem), como algum do GitHub, por exemplo, para sua máquina.

Para `clonarmos` o repositório basta acessá-lo na própria página do GitHub e clicar no botão verde escrito **clone or download**. Depois disso basta copiar o link que vai aparecer.

Agora, abra o terminal e vá até o local onde quer clonar o repositório e execute o comando `git clone link-copiado` e pronto, agora você tem esse repositório no seu computador. Parabéns!

Você pode editar os arquivos desse repositório como quiser, mas é mais seguro você criar uma ramificação (também conhecida como *branch*) dele. Mas o que é uma *branch mesmo*? Branch é uma versão independente e editável que se cria de um código. Um código pode ter várias branches, sem limite de números. A Branch principal do código, a "oficial" onde ele todo consolidado, é a master.

Enfim, primeiro vamos verificar em qual *branch* estamos pelo `git branch`. Logo, vamos criar uma nova *branch* pelo comando `git branch nome-da-nova-branch`. Para começarmos a usar essa *branch* utilizamos o comando `git checkout nome-da-nova-branch`.

Agora que já criou e acessou a nova *branch*, você pode fazer mudanças nos arquivos existentes, ou até criar e deletar outros. Depois de fazer alguma mudança, adicione ela ao index. Agora, o que é o index? Index é como se fosse um aeroporto onde você coloca as alterações feitas no código. Fazemos isso usando o comando `git add ..` Depois de colocar as alterações no Index, vamos

definir elas no repositório pelo `git commit -m "mensagem que descreve as alterações"`. Agora é só empurrar essas alterações para o repositório remoto por meio do comando `git push`.

Se alguém fez alguma alteração, vale a pena, antes de você mandar a sua, colocar essa alteração da outra pessoa no seu repositório local por meio do `git pull`, uma vez que caso não se faça isso, os códigos podem conflitar.

Agora, volte para o site do GitHub, vá até o repositório e lá aperte o **botão compare and pull request**, ou vá até a aba Pull Request e crie um clicando em **new pull request**. Assim, você está falando para pessoa hospedeira do repositório que ela pode, se concordar com as suas alterações, "puxar" essas modificações. Curiosidade: Se você quiser fazer isso, apenas vá ao pull request e clique em **merge pull request**. Mas só é possível fazer isso se você for dono do repositório ou possuir as permissões adequadas para tal.

Roteiro 4

Agora faça uma simulação de projeto no GitHub. Basicamente, clone um repositório, crie uma branch, faça algumas alterações e push. Depois disso, abra um Pull Request.

<https://github.com/tryber/sd-01-week4-5-project-meme-generator>

Por hoje é só pessoal! Até!