

Aula 2

Git vs. Github

- O git é a ferramenta que gerencia as versões e colaborações em projetos
- O Github é um serviço cloud que permite armazenar os projetos
- Existem outros, como Bitbucket e Gitlab. Todos usam a mesma ferramenta, o git.

Github

- O projeto que está na nossa máquina chamados de repositório (ou repo) do git local
- O projeto que está no github, chamados de repositório (ou repo) do git remoto

COMANDOS/ SALVANDO LOCALMENTE

SALVANDO LOCALMENTE

● git status

- Indica o status do repositório
 - Arquivos/pastas criados
 - Arquivos/pastas modificados
 - Arquivos/pastas removidos

● git add nome-do-arquivo

- git add .



- Envia os arquivos modificados, removidos e criados para a Staging Area (que é local)
- Também podemos utilizar a opção `git add --all` para adicionar todos os arquivos do repositório;
- Ou a opção `git add .` para adicionar todos os arquivos da pasta onde você se encontra;

- **git commit -m "mensagem"**

- ☐ Demarca uma versão do seu projeto com os arquivos que estiverem na Staging Area
- ☐ A mensagem deve explicar as modificações, criações e deleções feitas
- ☐ Não esquecer do -m

■ Caso esqueça, você vai entrar em uma parte do terminal, que, para sair, você deve digitar: `esc`
`esc :q`

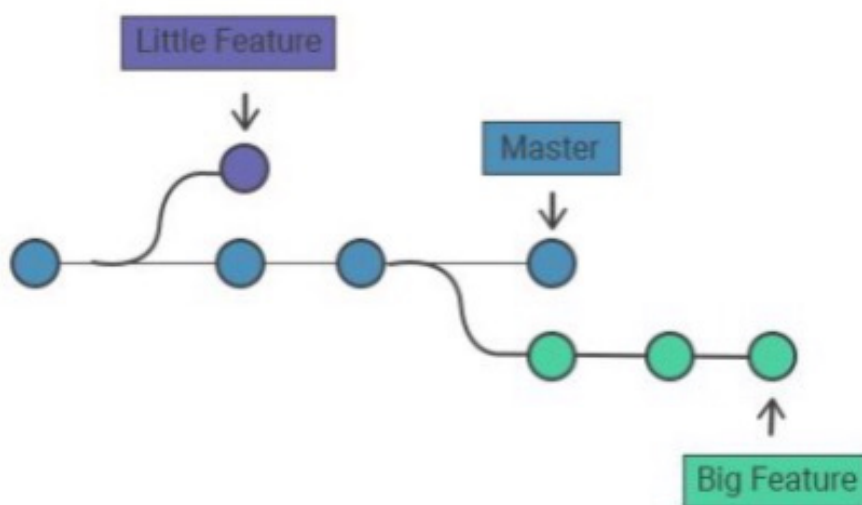
- Não esquecer das aspas ("")

- git log

- Permite verificar o histórico de commits do projeto

COMANDOS/DIVIDINDO O TRABALHO

- **git branch**



- ☐ Branch (ramo/galho) é uma ramificação do projeto principal
- ☐ Este comando em si mostra a lista de branches que estão no seu repositório local

☐ A branch padrão se chama main* e, a princípio, apenas ela vai existir no seu repositório

● **git branch nome-da-branch**

☐ Permite criar uma nova branch, com o nome que você escolheu

● **git checkout nome-da-branch**

☐ Permite acessar uma branch que já foi criada (localmente ou remota)

● **git checkout -b nome-da-branch**

☐ É uma junção dos comandos anteriores

☐ Ele cria uma nova branch e já acessa diretamente

COMANDOS/SALVANDO NO REMOTO

● **git push origin nome-da-branch**

☐ Envia as suas alterações feitas para a branch no repositório remoto

☐ Ele só envia as alterações que foram colocadas no commit

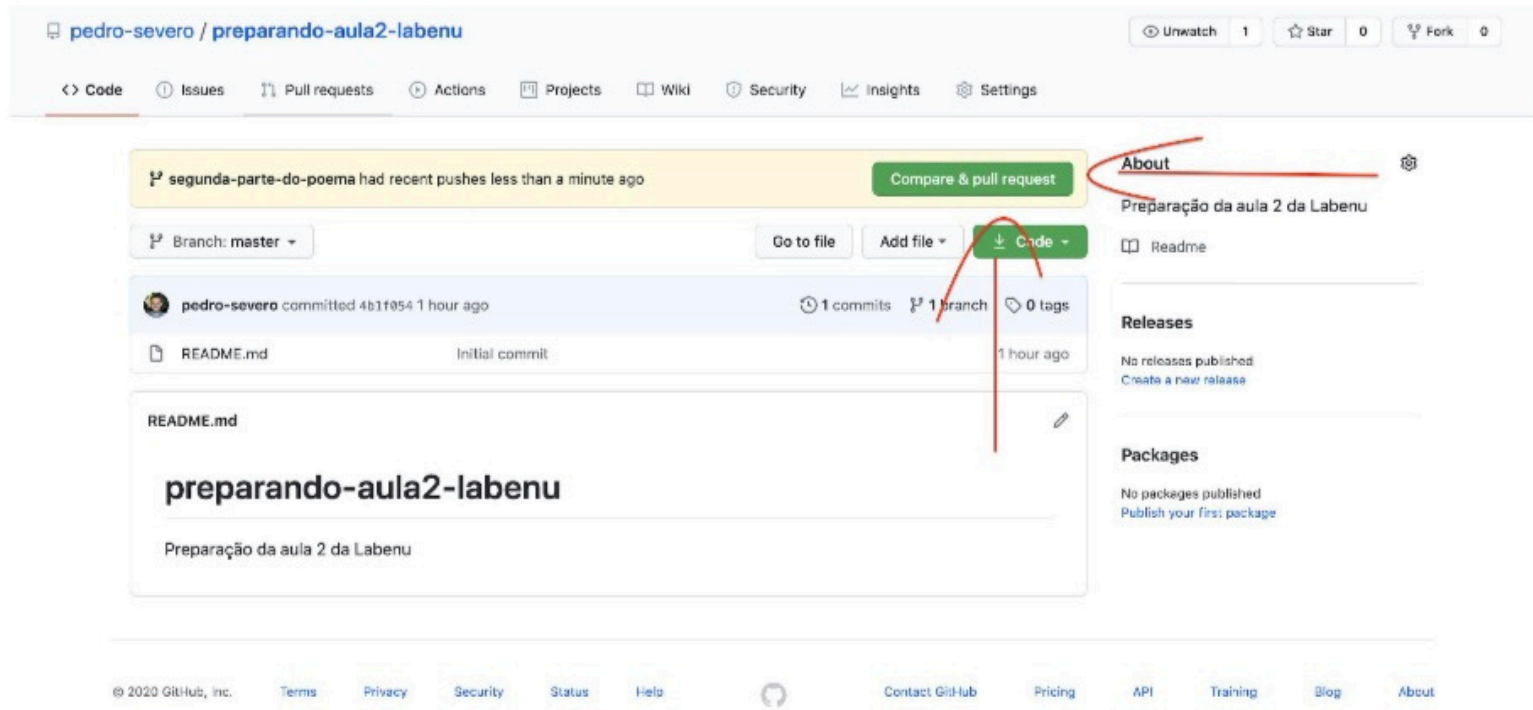
PR

> Pull Request (PR)

● Depois de fazer todas as alterações na sua branch, você deve querer que elas sejam mescladas com a branch principal (a master)

● A esta mesclagem, damos o nome de merge

● Para fazer um merge no GitHub, nós devemos criar um Pull Request (ou PR) antes



- Quando trabalhamos em equipe, os membros dela avaliam os nossos PRs
- Pedindo correções no código
- Sugerindo alterações
- Após o processo de Code Review (CR); e o seu código estiver aprovado, ele pode ser mergeado na main

Comandos/Atualizando o local

- `git pull origin nome-da-branch`
- Atualiza a branch em questão no seu repositório local com as alterações commitadas na branch remota
- Se você já estiver acessando a branch que deseja atualizar, o comando pode ser reduzido a `git pull`

RESUMO

- Começando o repositório

☐ git clone link-do-repo

● **Salvando localmente**

☐ git status

☐ git add nome-do-arquivo

☐ git add .

☐ git commit -m "mensagem"

☐ git log

● **Dividindo o Trabalho**

☐ git branch

☐ git branch nome-da-branch

☐ git checkout nome-da-branch

☐ git checkout -b nome-da-branch

● **Salvando no Remoto**

☐ git push origin nome-da-branch

☐ git pull origin nome-da-branch

● **Sempre queremos que as alterações de uma branch nossa sejam mescladas com as informações que já estão na master (merge)**

☐ **Para isso , devemos criar um PR**

☐ Solicitando aos nossos colegas de trabalho que avaliem o nosso código, dando sugestões de melhoria

● **Importante:** comandos de git não são o mesmo que comandos do terminal!

☐ Ex: git mkdir

● **Importante 2:** branch não é pasta!

Resumo

