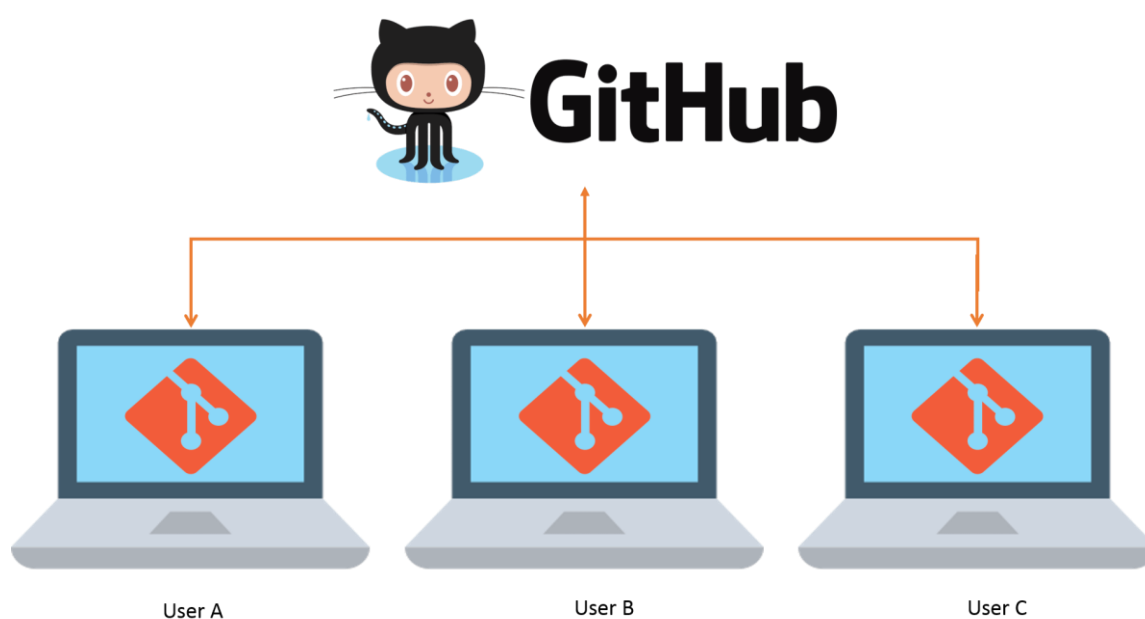


GIT & GITHUB



COMMANDS

1. `git --version`
 2. `git init`
 3. `rm -rf .git`
 4. `git config --global user.name "Emanuel Quintino"`
 5. `git config --global user.email "emanuelquintino@hotmail.com"`
 6. `git config --global core.editor "code -w"`
 7. `git config --global --unset use.email`
 8. `git config --list --global`
 9. `git config --list --local`
 10. `git status` (Working Directory, Stage Area, Repository(Logs))
 11. `git add .`
 12. `git rm --cached` (add file name)
 13. `git restore .`
 14. `git restore --staged .`
 15. `git commit -m "first commit"`
 16. `git commit --amend -m "new message"`
 17. `git reset --soft HEAD~1`
 18. `git reset --hard HEAD~2`
 19. `git log`
 20. `git log -n 3`
 21. `git diff`
 22. `git branch -M main`
 23. `git remote add origin` (add address)
 24. `git remote --v` (consult remote address)
 25. `git push -u origin main`
 26. `git push`
 27. `git pull`
 28. `git clone` (add link)
 29. `git branch`
 30. `git branch` (branch name)
 31. `git branch -D` (branch name)
 32. `git checkout` (branch name)
 33. `git checkout -b` (branch name)
 34. `git merge` (branch name)
 35. `git push origin` (branch name)
 36. `git push -u origin` (branch name)
 37. `git fetch`
 38. `git rm -r --cached .` (clear cached Git and remove file GitHub)
- **U** (Untracked): São os arquivos que ainda não são monitorados pelo GIT.
 - **M** (Modified): Arquivos commitados e agora foram modificados.
 - **A** (Added): Arquivos que não foram commitados mas já rodou git add.
 - **D** (Deleted): São arquivo que foram removidos.

Configurando SSH

O SSH é uma sigla para Secure Shell, um termo e protocolo de um mecanismo de segurança na rede.

Gerando a chave SSH

1. Abra Terminal (macOS / Linux) Git Bash (Windows)
2. Cole o comando abaixo, substituindo o endereço de e-mail pelo seu GitHub.
 - `ssh-keygen -t ed25519 -C "seuemail@gmail.com"`
3. Isto cria uma nova chave SSH, usando o nome de e-mail fornecido como uma etiqueta.
 - Generating public/id_algorithm key pair.
4. Quando aparecer a solicitação "Enter a file in which to save the key", pressione **ENTER**.

O local padrão do arquivo será aceito.

- Enter a file in which to save the key (/Users/you/.ssh/id_algorithm):
[Press enter]
5. (Opcional) Digite uma senha secreta segura no prompt. Para obter mais informações, consulte (<https://docs.github.com/pt/articles/working-with-ssh-key-passphrases>).
 - Enter passphrase (empty for no passphrase):[Type a passphrase]
 - Enter same passphrase again:[Type passphrase again]
 6. Inicie o ssh-agent em segundo plano
 - `eval "$(ssh-agent -s)"`
 7. Adicione sua chave SSH privada ao ssh-agent
 - `ssh-add ~/.ssh/id_ed25519`

Importando para o Github

1. Ainda no terminal, navegue até a pasta onde a chave SSH foi salva com o comando:
 - `nano ~/.ssh/id_ed25519.pub`
2. Copie o código inteiro que aparecerá no seu terminal e depois pressione CTRL + X para sair
 - A estrutura será algo parecido com o exemplo:
ssh-ed25519
AAbiroAACENzaC1lZDI1biroNTE5BBBBILkGTLmPXIpbiroT56Xh
JwRjUrU9C4osFEVYX1Ezys3fe33 oi@email.com.br
3. Acesse as **configurações** do seu Github
4. Clique em **SSH and GPG Keys**.
5. Clique no botão para criar uma **nova chave SSH**
6. Adicione um título para sua chave e cole o conteúdo copiado do passo anterior no campo **Key** e clique em **Add SSH Key**.
7. Agora sempre que for fazer um ``git clone``, utilize a opção SSH

