

Manual de Comandos Git

Autoria: Valmir Junior

- 1- git pull “endereço do repositório”** (traz as alterações remotas para sua máquina).
- 2- git push origin main** (envia as alterações da sua máquina, para repositório remoto).
- 3- git log** (ver o histórico de commits)
- 4- git rebase main** (atualiza sua branch com a main ou master).
- 5- git clone “endereço do repositório”** (para trazer um repositório remoto, para sua máquina).
- 6- git status** (mostra as alterações)
- 7- git commit -m “Título do commit”** (Para fazer um commit)
- 8- git commit –amend –no-edit** (Envia as alterações para o último commit já criado)
- 9- git branch** (lista todas as suas branch derivada da master).
- 10- git checkout -b “nome da branch”** (cria uma nova branch).
- 11- git branch -D “nome da branch”** (deleta uma branch).
- 12- git checkout “nome da branch”** (para troca de uma branch para outra).
- 13- git branch -m “Novo nome da branch”** (para renomear uma branch).

14- git help (lista todos os comandos git).

15- git help “nome do comando git” (Manual do comando git especificado).

16- git rm —cached “nome do arquivo” (Para remover um arquivo adicionado).

17- git restore “caminho do arquivo” (Para descartar mudanças).

18- git diff (Mostra todas as alterações realizadas no repositório).

19- git diff —staged (Mostra as alterações realizadas no commit).

20- git rm “nome do arquivo” (Para remover arquivo).

21- git mv “nome antigo do arquivo” “novo nome do arquivo” (Para renomear arquivos).

22- git revert HEAD~”número em que ocorreu o commit” (Para reverter um commit).

23- git log —oneline (Lista todos os commit de forma resumida).

24- git revert “id do commit” (Para reverter um commit).

25- git commit “nome do arquivo alterado” - m “mensagem do commit” (Assim você faz commit, com o git add incluso dentro).

26- git restore —source “rash do commit” . (Para voltar no tempo a um momento específico do código).

27- git log --author="nome de usuário do github" (Para ver commit de um mesmo autor).

28- git log -p (Informações detalhadas alteração de commit).

29- git switch "nome da branch" (Para mudar de branch).

30- git merge "nome da branch" (Para unir a desejada branch com a atual).

31- git stash (Ele cria um *backup* das modificações de seus arquivos.)

32- git stash list (consultar todos os *stash's* criados)

33- git stash pop (remover o primeiro stash da stack-pilha).

34- git stash clear (Realiza a remoção de todos os *stash's*).

35- git stash show (Lista todos os arquivos modificados no último *stash*).

36- git restore --staged "caminho do arquivo" (Para desfazer de um arquivo adicionado)

37 - git reset HEAD^1 (Apaga o ultimo commit).

38 - git fetch "nome do user do github da pessoa/copia chave ssh da branch" "nome da branch que deseja baixar":"nome para atribuir na branch na sua máquina" (Para baixar uma branch-PR).

39 - git commit --amend -m "Nova mensagem do commit" (Altera mensagem do ultimo commit).

40 - git rebase -i HEAD~(escolha a quantidade de commits) - {

40.1 Digite a letra i (para editar dentro do rebase).

40.1.1 Substitua a hash do commit para edit (para alterar um commit específico).

40.1.2 Substitua a hash do commit para drop (para excluir um commit específico).

40.2 Clique na tecla esc (Para salvar suas alterações dentro do rebase)

40.3 Digite wq (Para fechar a tela de rebase).

41 - git reset --soft "hash do commit" (Traz as alterações do commit, através do git add . você pode atualizar - Logo com quando commit e enviar "push" ele irá apagar o commit anterior, e só permanecerá os atuais)

42- git cherry-pick "hash do commit" (Para pegar um commit específico e colocar dentro da branch em que você está).

43- git push "nome do remote da pessoa que deseja enviar" "nome da sua branch": "nome da branch em que deseja enviar" (Para enviar algo para pull request de outra pessoa)

44- git remote add "ssh do repositório da pessoa" (Para adicionar o remote)

45- git remote -rm "nome do remote da pessoa" (Para excluir remote de outra pessoa)

46- git remote -v (Usado para listar todos os remotos, que você tem na sua máquina)

47- git push origin --delete "nomeBranch" (Para excluir a branch do remoto)

48 - git log -grep "numeroLPD" (Para filtrar todos os commits o qual possuem essa LPD)