Roteiro Prático Git

Instalação

1. Baixe e instale o Git: http://www.git-scm.com/
2. Configuração da conta:
   a. git config --global user.name "seuNomeDeUsuario"
   b. git config --global user.email meu_nome@email.com
3. Cmd
4. Cd modulogit
5. git init
6. git config –list

------- **Primeiro projeto** -----

1. git branch (são versões do sistema)
2. criar o readme e escreve algo nele "meu primeiro sistema no git"
3. git add –A
4. git commit -m "Primeiro Commit"
5. alteração no readme, criar um novo arquivo html, e um css
6. git add -A (só está idenficando pra registrar eu preciso comitar)
7. git status
8. git commit -m "Criando arquivos principais, index e style, bem como alteração do readme"
9. git log (vê tudo o que foi feito)
10. alterar o arquivo readme e criar outro index.html
11. git status
12. git add -A (adiciona todos os arquivos modificados)
13. git add index.html (adiciona o arquivo específico)
14. git status
15. git commit -m ou -am "Adiciona o index e altera o readme"
16. git log
17. git branch

------- **Recuperando versão** -----

1. git status e copia o código dele
2. git reset --soft (volta com tudo alterado sem comitar acesso antes do comit)
3. git reset --hard (volta com tudo e apaga os arquivos)
4. git status
5. git branch
6. cria um novo arquivo css
7. git commit -am "Criando um arquivo CSS"
8. git log
9. git reset --soft bota o código do que quer
10. git status

--- faz as alterações que quiser e corrije o bug e faz novo commit --

11. git commit -am "alteração no css"
12. git log

------- **Usando várias branchs** -----

1. git branch
2. git log
3. git branch nomedonovobranch (cria nova branch)
4. git branch
5. git checkout nomedonovobranch
6. altera o readme cria um novo arquivo
7. git status
8. git commit -am "Testando o novo branch"
9. git log

--- quando criamos um novo branch ele copia os arquivo e começa uma versão a partir disso---

10. git checkout branch (voltar pra master)
11. git log (não está o commit de antes)
12. git checkout branch (voltar pra branch criada de teste)
13. git status

--- tem um histórico diferente nos dois ---

14. alterar o readme usando a branch master
15. git commit -m “alteração no readme”
16. git checkout teste
17. abre o arquivo (tenho versões diferentes do meu sistema)
18. cria nova branch faz anteração e volta no outro

------- **Verificar diferenças entre arquivos** -----

1. git status
2. faz alterações e salva
3. git status (diz que tem arquivos modificados)
4. git diff (mostra as alterações)
5. acrescenta mais uma linha no readme
6. git diff
7. zerar o arquivo
8. git diff
9. Mantém a linha
10. git diff --name-only (mostra somente um arquivo)
11. adiciona nova linha no style.css
12. git diff style.css
13. git commit -am "nova linha adicionada no style.css"

----- se não quer comitar quer voltar as alterações ----

14. git checkout HEAD -- style.css (volta pro style.css antigo) o head (pega do branch que está no momento)
15. git diff

----- se quiser combinar os arquivos de outra branch ----

16. merge master

------- **Repositórios local e remoto** -----

1. Criar conta no github
2. Criar novo repositório
3. Gerar uma chave
4. git bash (entra no terminal)
5. ssh-keygen -t rsa -b 4096 -C angelamazzonettofw@gmail.com
6. aperta enter
7. vai na pasta ssh
8. tem o arquivo com chave publica e a privada
9. copia o arquivo - setting - new (isso vai permitir o envio do acesso)
10. git remote add origin https://github.com/AngelaMazzonetto/modulogit.git
11. git remote
12. git remote -v

---- fetch é a capacidade de pegar do repositório remoto e enviar pro local ---

---- push levar as coisas do meu repositório local pro remoto ---

13. git push -u orign master
14. alterar o readme e cria um novo arquivo
15. git commit -am "Atualizando readme mais uma vez"
16. git push origin master (empurrar pra nuvem o branch)
17. criar um index.html
18. git add (adicionar o index)
19. git commit -am "inserindo o index"
20. git push origin master (empurrar pra nuvem o branch)

--- ignorando arquivos ---

21. criar um novo arquivo .sql e uma pasta qualquer
22. .gitignore coloca os nomes dos arquivos que quer que ignore
23. *.sql ou qualquer/
24. git add – A
25. git commit -am "Adicionando o git ignore"
26. git push origin master

---- git revert ---

1. altera o readme
2. git status
3. git add e git commit
4. git log
5. pega o código do código que deu problema
6. git revert --no--edit código (volta)

---- git detele ---

1. git branch
2. git chckout teste
3. git push origin teste
4. atualiza o repositório
5. quero deletar o branch
6. git push origin :desenvolvedor
7. deletar branch local
8. git branch -d desenvolvedor

--- git pull ---

1. git push origin master (enviei as atualizações)
2. git status (ve que tá tudo atualizado)
3. novo arquivo via browser
4. git status tem coisas novas no remoto
5. git pull origin main (antes de dar um push dá um pull)

--- git clone ---

-- clonar projeto - puxar pro meu repositório local ---

- git clone url

------- **Contribuir com outro projeto** -----

1. Abrir o repositório online do projeto
2. Clicar em “fork”
3. git clone url
4. altera os arquivos
5. entrar no repositório
6. git add –A
7. git commit –m “Alteração do arquivo xxxxx”
8. git status
9. git branch
10. git remote –v

11. git push origin master
12. pull request