

Git (exercícios)

1. Qual o comando para obter a versão instalada do Git?

`git --version`

2. Qual o efeito da execução de cada um dos comandos abaixo?

- a. `git config -l`

Mostra o username e password

- b. `git mv a.txt b.txt`

Mover e renomear

- c. `git reset --hard`

Funciona como se fosse um revert, você deve setar qual branch deseja

- d. `git log -27`

Mostra os logs de commits

- e. `git help`

Mostra os comandos de git mais usados

- f. `git help reset`

Mostra ajuda referente ao command reset

- g. `git add --all`

Mesmo que o git add -A

- h. `git add -u`

Adiciona arquivos que foram modificados e que já estão no projeto

3. O fluxo “clássico” de interação com o Git é algo como “alterar um ou mais arquivos”, “acrescentar essas mudanças para serem contemplados no próximo commit” e, finalmente, executar um “commit”. Quais os comandos necessários para realizar os dois últimos “passos” desse fluxo?

`git add -u`

`git push`

4. Qual o comando deve ser executado para identificar o que foi alterado desde o último “commit”?

`git diff`

5. Em um dado repositório, arquivos simplesmente copiados para lá, ou seja, *untracked*, podem ser exibidos/identificados com que comando?

`git ls-files`

6. Qual o comando para efetuar um *commit*?

`git commit`

7. Qual o comando que devemos empregar para descartar mudanças ocorridas no arquivo teste.txt, por exemplo?

`git checkout HEAD -- teste.txt`

8. O que deve ser feito para que um determinado diretório do seu repositório seja ignorado pelo Git? Faça uma busca por **.gitignore**.

Adicionar o nome do diretório no arquivo `.gitignore`.

9. O que acontece se o seu repositório local for acidentalmente removido?

Ele é removido localmente, eu posso clona-lo novamente.

10. Como clonar um repositório remoto?

`git clone <url_repositorio>`

11. Em alguns cenários **git log** pode produzir extensos resultados. Se houver interesse em visualizar o histórico de um repositório, onde cada mudança é fornecida exatamente em uma única linha, qual o comando que deve ser empregado?

`git log -- path/to/folder`

12. Em qual arquivo o Git armazena informações de configuração empregadas por usuário?

No arquivo `config`.

13. Qual o comando para criar um repositório local?

`git init`

14. Qual o nome do diretório criado pelo Git quando se executa o comando **git init**?

`.git`

15. Qual o comando para adicionar todos os arquivos modificados? (Aqueles para os quais **git status** identificam como **modified**?)

git add -u

16. O Git faz uso do valor de *hash* conhecido por SHA1. O que isto significa? Qual o propósito? O que é SHA1?

SHA1 é um hash. O git faz uso desse hash pra verificar se houve mudança no arquivo.

17. Qual a palavra para indicar o último *commit* em vez do valor de *hash* SHA1 correspondente?

git log -n

18. Quando se cria dois arquivos usando um editor de texto qualquer e, na sequência, executamos o comando **git add -u**, os dois arquivos criados passam de *untracked* para *new file*?

Não, -u é usado para arquivos *trackeds*.

19. Qual o efeito da execução dos dois comandos abaixo, nesta ordem, em um dado repositório?

git reset --soft HEAD~1

Com este comando o último commit vai ser desfeito, mas você não irá perder as modificações dos arquivos.

git reset --hard

Com este comando o último commit é desfeito e você perde as modificações nos arquivos.

20. Após o emprego de um ambiente integrado de desenvolvimento (IDE), é comum a criação de arquivos e diretórios. Qual o comando que podemos empregar para remover arquivos e diretórios *untracked*?

git clean -f

21. Qual o nome do arquivo no qual podemos inserir a indicação para o Git de arquivos e diretórios a serem ignorados?

.gitignore

22. Quando se cria o arquivo *MinhaClasse.class* em um dado diretório e desejamos que o Git ignore não apenas este, mas arquivos *.class* em geral, por todos os membros de uma equipe que estão contribuindo com um dado projeto?

Adiciona a extensão do arquivo no arquivo .gitignore ou executa o seguinte comando

```
git rm -r --cached *.class
```

23. jQuery é uma famosa biblioteca em JavaScript. Consulte detalhes em <http://jquery.com>. O repositório Git correspondente encontra-se disponível em <https://github.com/jquery/jquery.git>. Faça o clone deste repositório **jqueryrepo**.

```
git clone https://github.com/jquery/jquery.git
```

24. No repositório **jqueryrepo**, criado no passo anterior, qual o efeito do comando **git shortlog -sne**?

Mostra a quantidade de commits juntamente com nome do usuário e email.

25. No repositório **jqueryrepo**, qual o efeito de **git remote -v**?

Mostra a url do repositório remoto

26. Um repositório Git pode ser etiquetado ao longo do tempo. Ou seja, *commits* específicos podem ser “marcados” ou “etiquetados” para facilitar referências posteriores. Para listar todas as “etiquetas” (*tags*) estabelecidas para um dado repositório, qual comando deve ser executado?

```
git tag
```

27. Caso um dado repositório retorne muitas “marcas” ou “etiquetas” para o comando **git tag**, como retornar apenas aquelas que atendem a determinado padrão, por exemplo, iniciadas por 2.0?

```
git tag -l 2.0
```

28. Qual o efeito do comando **git tag -a 3.4-gold -m “minha versão ouro”**?

O comando da forma que está apresenta o erro "fatal: too many params"

29. Após executado o comando acima, qual o efeito de **git show 3.4-gold**?

Se o comando anterior não tivesse dado erro, mostraria “minha versão ouro”.

30. O que o comando **git push origin 3.4-gold** teria como efeito?

Enviaria pro repositório remoto a versão 3.4-gold.

31. Após executar um *commit*, qual o efeito de **git commit --amend**?

Muda apenas a mensagem de commit.

32. Após executar **git add x.txt**, qual o efeito de **git reset HEAD x.txt**?

Marca o arquivo como untraked novamente.

33. Após alterar o conteúdo de um arquivo *committed* em passo anterior, qual o efeito do comando **git checkout -- a.txt**?

Sincroniza o arquivo a.txt com o repositório remoto.

34. Qual a diferença entre os comandos **git reset HEAD x.txt** e **git checkout -- a.txt**?

git reset HEAD x.txt apenas dá update no index, movendo o HEAD.

git checkout -- a.txt dá update no working tree.