|  |  |
| --- | --- |
| **Leitura de Software**  **Docente: Prof. Fábio Nogueira de Lucena**  **Discente: Alexandre José Messias de Lara**  **Matrícula: 201206315** |  |

# **Construção de software**

# **Git (exercícios)**

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

git --version

1. **Qual o efeito da execução de cada um dos comandos abaixo?**
   1. **git config -l:** Lista todas as variáveis configuradas no arquivo de configuração, junto com seus respectivos valores.
   2. **git mv a.txt b.txt:** Como o diretório de destino é o mesmo, ele simplesmente renomeia a.txt para b.txt
   3. **git reset --hard:** Reinicia o índice e a working tree. Caso alterações tiverem sido feitas após o commit, elas serão descartadas.
   4. **git log -27:** Mostra os logs de commits com uma limitação de no máximo 27 commits.
   5. **git help:** Este comando faz com que informações de ajuda sobre git sejam mostradas.
   6. **git help reset:** Mostrar informações de ajuda sobre o commando “git reset”.
   7. **git add --all:** Adiciona os conteúdos de arquivo ao índice, atualizando o índice não somente da working tree que possua o arquivo correspondente, mas também onde o índice já possui uma entrada. Este comando adiciona, modifica e remove entradas de índices para coincidir com a working tree. Como nenhuma especificação de caminho foi dada, todos os arquivos da árvore de trabalho serão atualizados.
   8. **git add -u:** Atualiza o índice onde possui uma entrada que combina com o caminho passado. Como nenhum caminho foi especificado, esse comando irá atualizar todos os arquivos rastreados da working tree.
2. **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?**

Primeiro: git add [file]

Segundo: git commit -m “mensagem de commit”

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

git diff

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

git ls-files . --exclude-standard --others

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

git commit -m “Mensagem”

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

git reset teste.txt

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

Crie ou edite o arquivo .gitignore e adicione o nome do diretório em uma linha do arquivo.

1. **O que acontece se o seu repositório local for acidentalmente cd removido?**

Todos os arquivos do meu repositório local serão removidos, porém as mudanças não reflitirão no repositório remoto, a não que seja realizado um commit. É possível reverter a exclusão por meio dos comandos git reset e git checkout, que podem ser combinados para obter o resultado desejado.

1. **Como clonar um repositório remoto?**

Através do comando ‘git clone’, exemplo:

‘git clone https://github.com/*seu-nome*/*seu-repositorio’*‘

1. **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 -1

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

No arquivo .gitconfig

1. **Qual o comando para criar um repositório local?**

git init [nome do repositório]

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

.git

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

git add -u

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

O git usa SHA com o propósito de segurança e de identificação de alterações nos arquivos. SHA1 é um algoritmo que recebe alguns dados como entrada e gera uma string única de 40 caracteres através desses dados. Ou seja, nenhuma outra entrada de dados deveria produzir o mesmo hash. Porém, caso a mesma entrada de dados seja usada, ela sempre irá produzir o mesmo hash.

Isso é extremamente importante, pois o número de revisão do git é nessa hash SHA1, então mesmo que várias pessoas estejam usando o mesmo repositório, se o número de revisão for igual para todas elas, isso significa que todos estão trabalhando naquele mesmo conjunto de arquivos. Isso traz maior segurança, pois se qualquer alteração for realizada em qualquer bit que seja, ou mesmo durante a transferência dos arquivos, se ocorrer qualquer perda de bits, o git irá identificar.

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

Revisão

1. **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, a terminação ‘-u’ apenas modifica arquivos que já foram adicionados, com este comando, novos arquivos não serão adicionados.

1. **Qual o efeito da execução dos dois comandos abaixo, nesta ordem, em um dado repositório?  
   git reset --soft HEAD~1  
   git reset --hard**

O primeiro não irá mexer com o arquivo index ou com a working tree, porém irá reiniciar a head para o commit que for determinado. Todos os arquivos modificados ficarão como mudanças a serem commitadas.

Já no segundo caso, o índice e a working tree serão resetados e quaisquer mudanças nos arquivos já rastreados desde o commit indicado serão descartadas.

1. **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 -fd

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

.gitignore

1. **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?**

Pergunta não está muito clara, mas caso queira ignorar todos os arquivos .class, deve-se colocar esta regra no arquivo .gitignore, a regra seria:

\*.class

1. **jQuery é uma famose biblioteca em JavaScript. Consulte detalhes em** [**http://jquery.com**](http://jquery.com)**. O repositório Git correspondente encontra-se disponível em** [**https://github.com/jquery/jquery.git**](https://github.com/jquery/jquery.git)**. Faça o clone deste repositório jqueryrepo.**

Realizado.

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

Retorna as mensagens de log da saída padrão, mostrando somente o resumo da contagem do commit, ordenadas pelo número de commits por autor e mostrando o nome e email de cada autor. Exemplo:

1714 John Resig <jeresig@gmail.com>

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

Mostra o endereço do repositório remoto.

origin https://github.com/jquery/jquery (fetch)

origin https://github.com/jquery/jquery (push)

1. **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

1. **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\*"

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

Cria uma annotated tag ‘3.4-gold’ com a mensagem de tagging “minha versão ouro”

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

É possível ver os dados da tag, assim como o commit que foi etiquetado:

tag 3.4-gold

Tagger: Alexandre Lara <thealexandrelara@gmail.com>

Date: Sun Aug 20 22:20:05 2017 -0300

minha versão ouro

commit b1b949d35ee995ee75c6968715a7f8c8ea601157

Author: Jordan Beland <jordan.beland@gmail.com>

Date: Tue Aug 15 22:07:49 2017 -0700

Tests: Update path calculation

Fixes gh-3756

Closes gh-3757

diff --git a/test/jquery.js b/test/jquery.js

index e337a79..d8eaf86 100644

--- a/test/jquery.js

+++ b/test/jquery.js

@@ -2,7 +2,8 @@

( function() {

/\* global loadTests: false \*/

- var path = window.location.pathname.split( "test" )[ 0 ],

+ var pathname = window.location.pathname,

+ path = pathname.slice( 0, pathname.lastIndexOf( "test" ) ),

QUnit = window.QUnit || parent.QUnit,

require = window.require || parent.require,

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

Iria atualizar o repositório remoto com o repositório local sob a tag 3.4-gold, caso houvesse permissão para isso.

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

O git commit --amend permite adicionar alterações ao último commit, por exemplo, na situação em que você esqueceu de adicionar alguma coisa ao último commit, seja um arquivo ou uma certa alteração aos arquivos.

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

As mudanças no arquivo e adição não serão commitadas.

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

Nenhum efeito porque o arquivo já foi commitado.

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

O primeiro irá resetar a HEAD para o estado identificado para o arquivo x.txt, enquanto o segundo irá restaurar o arquivo a.txt da working tree.