

## PROVA DE VERSIONAMENTO(COM RESPOSTA)

1. Durante o processo de commit, você percebe que adicionou um arquivo ao stage por engano com o comando `git add config.json`. Agora, você quer remover esse arquivo da staging area (sem apagar suas alterações no diretório de trabalho). Qual comando você deve usar?

- a) `git remove config.json`
- b) `git reset --hard config.json`
- c) `git restore --staged config.json`**
- d) `git checkout config.json`
- e) `git commit --amend config.json`

2. Qual comando exibe o histórico de commits no repositório?

- a) `git log`**
- b) `git history`
- c) `git status`
- d) `git commits`
- e) N.d.a

3. Como fazer o merge do branch feature no branch atual?

- a) `git integrate feature`
- b) `git merge feature`**
- c) `git rebase feature`
- d) `git apply feature`
- e) N.d.a

4. Você executa o seguinte comando no seu repositório Git:

**`git reset --soft HEAD~2`**

Qual é o efeito desse comando?

- a) Apaga os dois últimos commits e todas as alterações associadas a eles
- b) Volta dois commits no histórico e mantém as alterações no staging area**
- c) Descarta os dois últimos commits e mantém as alterações apenas no diretório de trabalho
- d) Reverte os dois últimos commits criando novos commits de reversão
- e) Remove os arquivos adicionados com `git add` nos dois últimos commits

5. Você executa o seguinte comando no seu repositório Git:

**`git reset --hard HEAD~2`**

O que esse comando faz?

- a) Move o ponteiro HEAD para dois commits atrás, mantendo todas as alterações no stage
- b) Deleta os dois últimos commits e envia as mudanças para o repositório remoto
- c) Volta dois commits no histórico, descartando todas as alterações no stage e no diretório de trabalho**
- d) Cria um novo branch com o estado de dois commits anteriores
- e) Reverte os dois últimos commits mantendo os arquivos modificados

6. Você está no branch develop, muda para o branch hotfix, e em seguida executa:  
git checkout -

O que acontece?

- a) Volta para o commit anterior no branch atual
- b) Apaga o branch hotfix e volta para o develop
- c) Volta para o último branch visitado, que no caso é o develop**
- d) Reverte o último commit no branch atual
- e) Sai do modo detached HEAD

7. Como você adiciona todos os arquivos modificados para o próximo commit?

- a) git add -u
- b) git add \*
- c) git add .**
- d) git stage all
- c) git add all

8. O que melhor descreve um “commit” no Git?

- a) Um backup completo do projeto em um determinado momento
- b) Um rascunho local com alterações não salvas no repositório
- c) Um ponto registrado no histórico com alterações confirmadas no repositório**
- d) Uma versão temporária criada para testes antes de publicar
- e) Um apontador para a última versão do código em um branch

9. O que é o Git e para que ele é usado?

- a) Uma ferramenta para editar e formatar código automaticamente
- b) Um sistema de controle de versão distribuído que rastreia mudanças no código-fonte**
- c) Uma plataforma para hospedagem de projetos de software na nuvem
- d) Um framework para automatizar testes de software
- e) Um sistema centralizado para gerenciar versões de arquivos

10. Você está trabalhando em um projeto Git e possui várias modificações em arquivos já rastreados, além de novos arquivos ainda não adicionados ao versionamento. Para salvar

temporariamente todas essas alterações incluindo os arquivos não rastreados e deixar seu diretório de trabalho limpo para trocar de branch, qual comando você deve usar?

- a) git stash
- b) git stash -u**
- c) git stash --keep-index
- d) git reset --hard
- e) git stash all