

 <b>FACULDADE DE VIÇOSA</b> Ampliando Horizontes		Sistemas de Informação	
<b>Disciplina:</b>	Tópicos Especiais 2 - DET476		
<b>Data</b>	23/03/2016	<b>Valor da Avaliação:</b>	20 pontos
<b>Professor:</b>	Vinícius Duarte Lopes		
<b>Aluno:</b>		<b>Matrícula:</b>	

FAÇA A PROVA ESCRITA PRIMEIRO, SÓ LIGUE O COMPUTADOR DEPOIS, PARA FAZER A ÚLTIMA QUESTÃO.

### Lista de comandos Git:

`git init` (*inicia repositório na pasta*)

`git add <arquivo1> <arquivo2> ... <arquivoN>` (*adiciona os arquivos do diretório atual*)

`git add .`

`git commit -m "Mensagem"` (*cria um novo commit no repositório*)

`git log` (*lista os commits do repositório*)

`git show HEAD` (*Mostra a diferença entre o último e o penúltimo commit*)

`git show <codigo-do-commit>` (*mostra a diferença entre o commit do código referenciado e o commit anterior*)

`git diff` (*mostra a diferença entre o diretório de trabalho e o último commit*)

`git checkout HEAD <arquivo>` (*faz arquivo no diretório de trabalho voltar ao estado do último commit*)

`git reset HEAD <arquivo>` (*retira o arquivo da Staging Area, Working Directory fica igual*)

`git reset <codigo-do-commit>` (*volta ao commit de código passado como parâmetro, descartando os commits que vieram depois*)

`git branch`

`git branch <novobranch>`

`git checkout <novobranch>` (*muda para o branch "novobranch"*)

`git checkout -b <novobranch>` (*cria o branch "novobranch" e muda para o branch "novobranch"*)

`git checkout <código do commit>` (*volta ao commit do código especificado, sem perder nenhum commit, Detached HEAD state*)

`git merge <nome_do_branch>`

`git clone <endereço>`

`git pull origin <branch>` (*atualiza o repositório local com os commits do repositório referenciado pelo link origin, no branch especificado como parâmetro*).

`git push origin <branch>`

`git remote add <nome_do_remote> <endereço>`

`git remote -v` (*mostra a lista de links remotos que foram configurados no repositório*)

### Questão 1 (5 pontos)

Complete as informações faltantes na lista de comandos acima. Ou seja, o que fazem os comandos:

- (A) `git add .`
- (B) `git branch`
- (C) `git branch meunovobranch`
- (D) `git merge meunovobranch`
- (E) `git remote add origin <endereço>`
- (F) `git push origin master`
- (G) `git clone`

### Questão 2 (3 pontos)

Explique o que faz a funcionalidade *Fork* do GitHub.

### Questão 3 (4 pontos)

Logo após criarmos um branch podemos mudar para o branch original, do qual o novo branch foi criado. Quando o novo branch fica em um estado em que o HEAD dele é um commit diferente do branch original, ainda pode ser que possamos trocar de branch. Porém, há uma situação em que o Git não nos deixa trocar do branch novo para o original. Explique essa situação.

### Questão 4 (8 pontos) APÓS TERMINAR A PROVA ESCRITA. ROTEIRO.

- (A) Crie uma pasta chamada prova01-det476.
- (B) Adicione dois arquivos .txt, chamados de arq1.txt e arq2.txt
- (C) No arq1.txt adicione o conteúdo abaixo:  
x  
y
- (D) No arq2.txt adicione qualquer conteúdo.
- (E) Inicie um repositório Git no diretório criado.
- (F) Faça o Git ignorar o arq2.txt.
- (G) Crie um commit nesse ponto, depois de adicionar todos os arquivos que não foram ignorados.
- (H) Adicione a seguinte linha no arq1.txt, ele deverá ficar com o seguinte conteúdo:  
x  
y  
z
- (I) Crie um commit nesse ponto.
- (J) Crie um branch, escolha um nome qualquer para ele.
- (K) Mude para o branch criado.
- (L) Mude o arq1.txt para o seguinte conteúdo:  
x  
7  
z
- (M) Faça um commit nesse ponto.
- (N) Mude para o branch master.
- (O) Mude o arq1.txt para o seguinte conteúdo:  
x  
1  
z
- (P) Faça um commit nesse ponto.
- (Q) Faça um merge com o branch que você criou.
- (R) Crie um novo commit que resolve o conflito, colocando um valor qualquer para resolver o conflito.