FACULDADE SENAC GOIÁS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Raffael Aparecido da Silva Monteiro

CONTROLE DE VERSÃO

QUESTIONÁRIO

GOIÂNIA- GO
2019

1. **Diferencie Git de Github.**

O que é o Git?

O Git é um sistema de controle de versão de arquivos. É um software livre e muito utilizado no desenvolvimento de software onde diversas pessoas estão contribuindo simultaneamente, podendo criar e editar arquivos. Sempre quando alguém disponibiliza sua parte do projeto no Git, ele gerencia as alterações feitas e guarda um histórico. Isso é importante pois se houver algum problema você pode desfazer as alterações e voltar para a versão que estava estável.

O que é o GitHub?

O GitHub é uma plataforma onde você pode armazenar seus projetos. É como se fosse uma rede social, só que de códigos, onde seus desenvolvedores podem disponibilizá-los para outras pessoas verem. Quando seu projeto está no GitHub, você pode facilmente baixar uma cópia em outro computador. É uma plataforma gratuita e armazena milhões de projetos, tanto open source, pessoais e até mesmo comerciais.

Diferença

A diferença entre git e github é que o git é só uma ferramenta para versionar projetos, enquanto o github é o site no qual você colocará esses projetos versionados. Uma analogia válida seria que o git é seu pincel e tintas enquanto o github é um museu.

2. **Descreva a função de cada um dos comandos abaixo:**

a) **Init**

Criar novo repositório:

git init

b) **Add**

Adicionar arquivo/diretório (staged area):

Adicionar um arquivo em específico

git add meu_arquivo.txt

Adicionar um diretório em específico

git add meu_diretorio

Adicionar todos os arquivos/diretórios

git add .

Adicionar um arquivo que esta listado no .gitignore (geral ou do repositório)

git add -f arquivo_no_gitignore.txt

c) Status

Verificar estado dos arquivos/diretórios:

git status

d) Commit

Comitar arquivo/diretório:

Comitar um arquivo

git commit meu_arquivo.txt

Comitar vários arquivos

git commit meu_arquivo.txt meu_outro_arquivo.txt

Comitar informando mensagem

git commit meuarquivo.txt -m "minha mensagem de commit"

e) Log

Visualizar hitórico:

Exibir histórico

git log

Exibir histórico com diff das duas últimas alterações

git log -p -2

Exibir resumo do histórico (hash completa, autor, data, comentário e qtde de alterações (+/-))

```
git log –stat
```

Exibir informações resumidas em uma linha (hash completa e comentário)

```
git log --pretty=oneline
```

Exibir histórico com formatação específica (hash abreviada, autor, data e comentário)

```
git log --pretty=format:"%h - %an, %ar : %s"
```

- %h: Abreviação do hash;
- %an: Nome do autor;
- %ar: Data;
- %s: Comentário.

Exibir histório de um arquivo específico

```
git log -- <caminho_do_arquivo>
```

Exibir histórico de um arquivo específico que contêm uma determinada palavra

```
git log --summary -S<palavra> [<caminho_do_arquivo>]
```

Exibir histórico modificação de um arquivo

```
git log --diff-filter=M -- <caminho_do_arquivo>
```

O pode ser substituido por: Adicionado (A), Copiado (C), Apagado (D), Modificado (M), Renomeado (R), entre outros.

Exibir histório de um determinado autor

```
git log --author=usuario
```