**FACULDADE SENAC GOIÁS**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**2º PERÍODO**

**LUANA FERREIRA DE ALMEIDA**

**QUESTIONÁRIO GITHUB**

**GOIÂNIA- GO**

**2019**

**1º) Diferencie Git de Github.**

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.

O GitHub é uma plataforma onde você pode armazenar seus projetos. É parecido com uma rede social, só que de diferente ela tem códigos, onde seus desenvolvedores podem disponibilizá-los para outras pessoas verem. A facilidade do seu projeto é que no GitHub, você pode facilmente baixar uma cópia em outro computador.

**Diferença:**

O que diferencia o GiteGitHub é que o Git se refere a uma ferramenta para versionar(versão) projetos, enquanto o GitHub é o site no qual você colocará esses projetos versionados.

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

1. **Init**

**Criar novo repositório:** git init

1. **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

1. **Status**

### Verificar estado dos arquivos/diretórios: git status

1. **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"

1. **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>

**Exibir histório de um determinado autor :** git log --author=usuario