**Pesquisa Git e Github**

RODOLFO FARIA DA SILVA JUNIOR

GUILHERME ENRIQUE LIMA DA SILVA

ETEC CIDADE TIRADENTES

O que é um Controle de versão bom isso é fácil de responder.

Controle de versão é uma pratica de rastrear e gerenciar as alterações em um código de Software cujo sua principal finalidade é de armazenar todo o histórico de desenvolvimento do documento, isso deste o primeiro envio ate a sua ultima versão, bastante útil para resgatar as versões antigas desse prometo para economizar o tempo ao invés de ficar desfazendo as alterações feita quando o assusto sai muito do controle.

**Qual é as diferenças entre o Git e GitHub?**  
Git e o GitHub ambos são um software de controle de versão mas o que podemos observar entre suas diferenças é que um é de uso para projetos individuais (Git) e o outro para projetos coletivos (GitHub).

Outra diferença que não envolve seu uso em si, mas sim seus direitos de imagem, o Git é um programa de código aberto, já o GitHub é propriedade da Microsoft;

E quando falamos de Git e GitHub, podemos notar uma certa diferença, sendo ela é que o Git é um controle de versão já o GitHub é um serviço de hospedagem mais como uma rede social da programação.

**Quais são os principais estados o repositório pode apresentar?**Branch e Master.

**Quais os principais comandos do Git? e o que eles fazem?**  
Bom como o Git como qualquer outro sistema de programação é claro que ele vai ter seus comandos chave sendo esses:

1- Git config: esse comando serve para novos usuários sendo esse fundamental para configurar sua identidade de usuário, inserindo informações como nome e Email.

EXEMPLO:  
$ git config –global user.name “Seu nome” - (para colocar seu nome)

$ git config –global user.email “Seu email” - (para colocar seu Email)

lembrando que caso queira subir seu código para o Github tanto o nome quanto o Email tem que bater com a sua conta do Github.

2-Git init: esse comando fará você criar um novo repositório que é o responsável por armazenar seus projetos.

EXEMPLO:  
$ git init

3-Git clone:"esse comando ira copiar um repositório existente"

EXEMPLO:  
git clone <URL do seu projeto> - (URL = Link do site).

4-Git add: esse comando ira adicionar o sua parte do trabalho no material original no caso seu repositório.

EXEMPLO:  
git add .

5-Git commit: esse comando adiciona uma mensagem no trabalho que você adiciono no Git deixando ela em uma fila.

EXEMPLO:  
$ git commit -m “seu comentário”

6-Git branch:"esse comando serve para você criar seu próprio caminho no repositório, pos é bem comum aver variações em seu repositório"

EXEMPLO:  
$ git branch (lista todas as ramificações)

$ git branch <nome\_do\_branch> (cria um branch com o nome especificado)

$ git branch -d <nome\_do\_branch> (deleta o branch com o nome especificado)

7-Git remote: O comando Git remote estabelece uma conexão entre seu repositório local e um repositório remoto.

EXEMPLO:  
$ git remote add <nomecurto> <url>

8-Git push:"esse comando serve para subir suas modificações para um repositório remoto conectado anteriormente com git remote"

EXEMPLO:  
$ git push -u <nome\_curto> <nome\_do\_branch>

**Como subir relacionar o repositório local com o remoto?**Isso parece complicado de começo, mas não é um bicho de sete cabeças tão grande assim, uma das maneiras de fazer isso é usando o Git Pull.

O Git pull funciona da seguinte maneira, nada mais nada menos ele é um comando usado para atualizar sua branch locais de acordo com as branchs remotas, o que são branchs você me pergunta? Simples branches são: branchs em outras palavras são Ramificações no Git.