**Introdução ao Git e ao GitHub:**

**O que é o Git?**

Git é um sistema de versionamento de código distribuído, ou seja, um sistema de controle de versão que é capaz de registrar qualquer alteração realizada em um determinado código, ele é distribuído pelo fato de poder ter vários repositórios autônomos e independentes, um para cada desenvolvedor. Utilizando de comandos específicos no terminal do Git é possível manter o repositório sempre atualizado em todas as máquinas ou restaurar versões anteriores.

Vale destacar que o Git foi desenvolvido para o versionamento de códigos, porém ele é capaz de fazer o controle de vários tipos de arquivos, por exemplo a extensão “.txt”.

**O que é GitHub?**

Muitos consideram o GitHub como a rede social dos programadores, lá é possível armazenar códigos e projetos em repositórios de modo que este sirva como repositório e sirva de aprendizado para outras pessoas que possam se inspirar em seu projeto, podemos afirmar que esta é uma plataforma para gerenciar seu código e criar um ambiente de colaboração entre desenvolvedores.

**Termos importantes relacionado ao Git e ao GitHub:**

-Repositório: é uma pasta onde colocaremos o nosso projeto, ele pode ser local (hospedado na sua máquina) ou pode ser remoto (hospedado no GitHub);

-Branch: é uma ramificação (branching) que você fará no seu projeto, deixar de atualizar a ramificação principal e criar uma nova ramificação, isso poderia ser feito no caso de uma equipe de programadores, cada um trabalhando em uma parte(começo, meio fim), trabalhando em Branch separadas que serão reunidas posteriormente na Branch main;

-Commit: é postar as alterações que você faz no seu projeto, isso pode ser feito na linha principal ou na branch, é importante dar commits sempre que for feita uma alteração que funcione no código, o commit não atualiza o repositório, mas mantém salva as alterações e espera o Push;

-Merge: é uma junção (merging) da branch alternativa com a linha principal;

OBS: Se não for alterada a mesma linha, o merge ocorre naturalmente, mas se for na mesma linha o Git pede para que o usuário confira e decida as mudanças.

-Remote: faz com que o repositório da sua máquina seja conectado com o repositório do GitHub;

-Push: é o ato de empurrar seu código para o repositório do GitHub;

-Pull: é oposto de push, você puxa o que está no GitHub para a sua máquina.

**Qual a diferença entre a GUI e a CLI:**

Ambas são interfaces de usuários.

GUI é a interface gráfica (Graphic User Interface) do Git, é um meio de utilização do Git que permite que o usuário exerça suas tarefas sem necessitar de conhecimento profundo acerca dos comandos. Acessamos ela pelo Git GUI.

CLI é a interface de código, encontrada em sistemas operacionais como o terminal do Linux e o cmd do Windows, nela é possível realizar ações não disponíveis na GUI por meio de comandos. Acessamos ela pelo Git Bash.

**O Git e o GitHub são interfaces seguras?**

O git é um sistema distribuído e seguro pois o código que vive lá é a versão mais recente do código, imagine que muitas pessoas usam o mesmo código, na máquina de todas as pessoas haverá a mesma versão

Todo o conteúdo dos códigos, a relação entre os arquivos, diretórios e versões são criptograficamente protegidos por um algoritmo chamado de SHA1. SHA (secure hash algorithm) algoritmo de hash seguro, conjunto de funções hash criptográficas projetadas pela NSA(agência de segurança nacional dos EUA), a encriptação gera um conjunto de caracteres identificador de 40 dígitos , é uma forma curta de representar um arquivo, a cada alteração uma nova chave de 40 dígitos é gerada. De forma simplificada, o SHA1 embaralha determinado arquivo, gera um conjunto de 40caracteres e embaralha eles, eles são sempre únicos

É garantido que não ocorram mudanças acidentais ou maliciosas, assegurando que o histórico de alterações seja completamente rastreável.

**Chaves SSH e Tokens ou HTTP:**

Para realizar operações no GitHub podemos escolher entre um dos 3 métodos, o HTTP é o meio padrão de manipulação dos arquivos, exige que o usuário faça login de seu usuário e senha, já o SSH dispensa isso, ele cria sua chave com criptografia a qual seria utilizada no lugar do login.