**O que é o git e qual seu objetivo?!**

É um sistema de controle de versão, tendo como objetivo o compartilhamento de código para ter acesso as todas as suas mudanças entre os colaboradores, além de evitar conflitos nas alterações realizadas. Também pode “socorrer” o desenvolvedor caso ele faça alguma alteração que haja conflito e dê erro no código, pois ele pode voltar e ter acesso a versão anterior a sua alteração.

**Onde podemos usar o git?**

Podemos usá-lo em todo projeto que tenha todo tipo de arquivo, seja código, imagem, texto, áudio...

**Vantagens e desvantagens do git:**

Flexibilidade, segurança - ele é protegido por um algoritmo chamado SHA1 que protege contra alterações maliciosas e pode rastrear o histórico de alterações; Desempenho.

Maior complexidade, já que exige que os devs tenham um conhecimento muito grande para fazer uso dele

**O que é um repositório git?**

É o local onde seus projetos ficarão armazenados, localmente ou remotamente, a gente tem github como exemplo de repositório remoto.

**O que são branches?**

Branch é tipo uma ramificação do código, é tipo para você ter segurança em fazer alteração, não mexer na fonte principal do seu código, tipo assim, eu tenho um ambiente que armazena um estoque de supermercado aí chegaram produtos novos e eu quero adicionar, mas para não comprometer o código principal eu faço isso a parte para ter certeza do funcionamento e aí sim posso colocar esse no meu projeto principal. É basicamente fazer novas implementações sem afetar o que está em funcionamento.

**O que são commits?**

Commit são basicamente o conjunto de alterações que você faz, ele guarda as alterações, uma mensagem que contém o que foi feito e quem a fez. Ele também te dá a possibilidade de voltar em alterações anteriores, em algum ponto específico do seu código.

**Alguns comandos:**

Eu criei uma pasta no meu desktop e iniciei o git bash para acessar a pasta com o comando cd “Repgit” que foi o nome que coloquei na pasta. Depois disso usei o git init para criar um repositório local na minha pasta.

**Git init:** serve para criar meu rep local, caso eu já tenha uma pasta criada posso criar o rep procurando-a pelo git no cmd.

**Git add**: criei um novo arquivo html e utilizei o comando git add \* para adicionar meu arquivo ao “projeto”.

**Git commit**: confirmei a criação do meu novo arquivo, não utilizei as mensagens, mas posso utilizar o comando de commit com um traço m (-m) no final e uma mensagem resumindo o que eu fiz.

**Git status:** verifica se houve alguma alteração na branch,

**Git log:** serve para ver quais ver quais commits já foram feitos, as mensagens adicionadas, o horário e a data. Além de ver o hash de cada alteração, que é tipo um ID.

Logo após criar meu repositório local e adicionar mais um arquivo html para fazer commit, criei um repositório no github para enviar o meu repo local para um remoto. Depois de criar o rep por lá, peguei a url que ele me dá e coloquei no terminal junto ao comando **git remote.** Depois disso, usei o comando **git push** para enviar o projeto do meu computador para o github.

**Git reset:** para desfazer alterações e voltar a algum commit anterior, só precisar usar o comando com o hash/id do commit a qual você quer voltar.

**Ls ou dir:** ver lista de itens que estão na pasta selecionada.

**CD ..:** volta para a pasta do desktop, ou qualquer outra pasta anterior a que você esteja.

**Git remote add origin + link do repositório:** mandar meus arquivos para o github/rep remoto.

**Git push origin –u master:** enviar meus arquivos definitivamente para o github.

**Quando é necessário adicionar mais arquivos ao repositório não precisa utilizar o comando “git remote add origin”, somente o git push.**