**Diferença entre Git e GitHub**

GitHub é a plataforma que irá armazenar os arquivos do projeto, já o git é a ferramenta que manipula os commits.

**Repositório**

O repositório é como se fosse a pasta de arquivos do github, nesse caso ficará hospedado fora do computador. No caso do git, quando ainda estamos manipulando localmente, devemos tomar cuidado para não apaga-la.

**Git Init – O que é e Cuidados**

Toda vez que formos inicializar um repositório, precisamos colocar o seguinte comando:

**> Git init**

No entanto, é preciso tomar cuidado com o seguinte, ao inicializar o repositório dentro de um repositório já existente temos que verificar que ele irá reinicializar o mesmo, podendo afetar o projeto. Assim, para evitar, execute o seguinte comando:

**> Git status**

Assim, em caso de erro, ele retorna que **NÃO EXISTE REPOSITÓRIO GIT**.

**Sincronizando Repositório (Como conectar Github e Git)**

Para realizar as configurações, precisamos primeiro configurar algumas variáveis globais:

git -config --**global** user.email "rodrigo.alura87@gmail.com"

git config --**global** user.name "Rodrigo Ferreira"

Por fim, após isso, vamos executar o seguinte comando:

ssh-keygen -t ed25519 -C "rodrigo.alura87@gmail.com"

No caso do email, precisa ser o mesmo usado pelo GitHub. Por fim, é só adicionar a chave ssh ao Github.

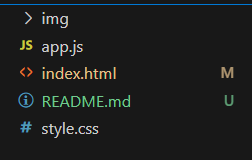
Aula02 – Colaborando em Projetos

(Clonando Repositório)

Para fazer o clone de um projeto, usamos:

git clone https://github.com/rodrigoalura87/numero-secreto.git

Quando trabalhamos com o VSCODE, ele nos sinaliza o status do arquivo, confira:



M: A letra M representa o estado Modified, do português modificado. Isso significa que o arquivo já existia no repositório, mas que recebeu alguma modificação que ainda não foi registrada no Git.

U: A letra U representa o estado Untracked, do português não rastreado. Isso significa que o arquivo ainda não existia no repositório e que ainda não teve seu registro (commit) feito no Git.

Enviando Commits de outras contas

Para aceitar commits e pushs de outras contas, devemos adicionar **colaboradores**. Após isso, podemos executar:

Git push origin main

Baixando novos commits:

Para fazermos isso, usaremos o git pull

Git pull origin main

Para verificar o resultado disso, use também o **git log**.

Reverter Alteracoes

Git Revert – cria um novo commit com as alterações revertidas.

Já para apagar um commit, utilizados o seguinte:

Git reset –hard commit

Alterando o Ultimo Commit

Git commit –ammend -m “mensagem”

Ignorando pastas/arquivos

Existe um recurso que nos permite criar um arquivo para informar ao Git quais diretórios e arquivos do projeto ele deverá ignorar. Esse arquivo é chamado de .gitignore. Podemos criar um arquivo com esse nome no nosso projeto.

No Visual Studio Code, criaremos um novo arquivo. O nome desse arquivo deve ser .gitignore, com um ponto no início, pois será um arquivo oculto. Após criar o arquivo .gitignore, incluiremos nele quais arquivos e diretórios do projeto queremos ignorar.

Por exemplo, vamos supor que temos uma pasta chamada temp/ no projeto, que é uma pasta temporária, com arquivos temporários. Estamos adicionando essa pasta no arquivo .gitgnore para que o Git ignore esse arquivo. Se tivermos outros arquivos e diretórios, cada um deles estaria separado por linhas.

Conhecendo o GIST

Permite compartilhar trecho de código sem criar um repo...