**Curso de Git e Github Alura**

md – Markdown é uma linguagem de estruturação para manipular texto

O markdown nos proporciona a possibilidade de configar do seu jeito ele aceita h1, #, >, <, ... ..., etc

É recomendado cria um readme em cada projeto por que podemos esquecer do que foi feito com o passar do tempo

**Dentro do github dentro do repoditório que vc quer editar etc, clicando 3 vezes no ponto(...) ela abrirá um editor vscode limitado para facilitar algum tipo de programação ou edição do seu trabalho, com limitações mais é um ótimo editor.**

Adicionando um colega para ajudar no desenvolvimento do trabalho:

Ir e Settings > colaboradores > perfil do github que quer adicionar e adicionar

Alguns comandos do Git:

git clone – Faz um clone do repositório do github para a sua maquina na pasta que você selecionou para fazer esse comando;

git clone <repositório> <meu-projeto-clone> - cria um clone em uma pasta definida por você

git clone –branch new-feature <repositório> - clonar o repositório a partir de uma branch especifica, diferente da original;

git log – permite visualizar todo o histórico tudo o que foi feito durante o tempo de programação nesse repositório;

git log –oneline – mostra todos os commits que foi feito de forma simplificada;

git log –p – Imprime as informações detalhada dos commits feitos;

git log –author=”nome do usuário” pesquisar as informações do autor daquele commit;

git log –since=1.monht.ago –until=1.day.ago – pesquisar informações por data no caso desse comando acima estamos buscando mudanças feitas do mês atrás ates um dia atrás;

git log –pretty=”formta:%h %s” – para formatar as informações de commit;

git pull – você consegue fazer a atualização sem clonar clonado pode apagar tudo o que tinha em seu repositório ele indica também se houve algum tipo de atualização no repositório feito por algum outro programador;

git status – Mostra tudo o que está de diferente no projeto se foi feito alguma modificação se precisar ser comitado etc;

git commit – Esse comando executa o commit que foi feito no código digitamos o (-m) para digitar uma mensagem em seguida com aspas –m “kjsfjafska”;

git push – O push empurra os commits para o github digitando origin main enviamos para divisão que estamos trabalhando que é nesse caso a divisão origin main;

Diff – mostra a linha que foi modificada

Retornar a versão anterior – dentro do github entrando o histórico você consegue visualizar as alterações que foram feitos copiar e usa-lá no git bash usando o comando

git restore –source “código copiado” . ou no próprio git bash com o comando git log –oneline ele vi trazer todos o histórico copie o código da versão que você quer voltar e execute o comando escrito acima

Obs: podemos utilizar o nome do arquivo modificado exemplo índex.html ou caso for feito modificações em mais de um arquivo só o ponto (.) já atualiza toda a modificação feita por boa pratica já melhor utilizar só o ponto

Feito isso agora é dar um push após isso será necessário dar um commit por que essa modificação foi feita apenas local temos que jogar na rede

git add – utilizado para adicionar arquivos novos ou mesmo até uma modificação feita;

Branch são ramificações do projeto

git chekout –b o nome da branch – ao clicar em enter ele cria a nova branch e já muda o estado para ela;

git branch – Conseguimos visualizar todas a branchs que temos;

git switch – Utilizado para selecionar uma branch local que queremos selecionar exemplo git switch novo codigo;

git fetch – alterar para um branch remoto ;

e seguida use o comando:

git switch –c <nome-do-branch-local> --track <remoto-origin>/<nome-do-branch-remoto>

Merge – mergiar mandar para a main após aprovação o código estiver tudo correto

Git merge (nome do repositório) – então pega esse repositório e envia para o repositório que você estiver

Então é só dar o push e então será feito a modificação

Git config - - local user.name ”nome” - para só o local que estou trabalhando ou

Git config - - global user.name “nome” para a máquina toda

Git init –bare – indica que esse repositório é puro, um repositório que só contem as alterações dos arquivos não contém a copia dos arquivos físicos permitindo adiconar esse repositório como remoto em outro

**git config core.bare true –** Caso não tenha executado o git init sem o bare use esse comando para transformar em bare