EXERCICIO-APLICACAO-GIT-CMD

O GITHUB já foi praticado em sala de aula.

Você reforçou os conceitos, assistindo ao vídeo do professor.

Agora, pratique o GIT CMD, integrado com o HUB, seguindo os passos descritos.

1º) No GITHUB, crie um Repositorio chamado ExemploGITcmd

2º) Suba um arquivo WORD com o nome EXEMPLO-APLICACAO-GIT-CMD. Coloque nesse arquivo o texto”Meus estudos de GIT”.

3º) Agora, você vai operar modificações no seu PC, criando Branch local de manutenção

Faça o Clone do repositório no seu PC local:

1. Execute o **GIT CMD** e quando a janela abrir...
2. Digite **git clone** *<URL do seu projeto>*
3. Uma pasta será criada em C:\users\*seunome\nomedoprojeto*

4º) Faça o login no Git remoto para poder fazer as manutenções, sincronizando dados:

1. **Git config –global user.email** <*seuemail>*

5º) Verifique se os arquivos foram clonados no seu PC. Execute dentro da pasta criada:

1. **Dir \*.\***

6º) Crie sua Branch de manutenção:

1. **Git branch** *manut01*
2. Execute em seguida, o seguinte comando para verificar as Branchs disponíveis:

**git branch**

7º) Altere o arquivo EXEMPLO-APLICACAO-GIT-CMD na Branch de manutenção, adicionando o texto “Meu primeiro exercício com GIT Command” e salve . Para fazer isso, execute os passos:

1. **Git checkout** *manut01*
2. *Verifique o status da modificação com* **git status**
3. SE VOCÊ TENTAR DAR UM CHECKOUT E VOLTAR PARA A MASTER O GIT AVISARÁ QUE EXISTE UM ARQUIVO ALTERADO NA BRANCH E NÃO COMITADO NA MASTER
4. Se se arrepender da mudança, digite **git checkout --** <nome do arquivo> e ele retornará a versão anterior (REVERT)

8º) Prepare as mudanças para o commit (coloque na situação de Stage), usando:

1. **git add** <nome do arquivo> OU
2. **git add .** que adiciona todos arquivos modificados no controle

9º) Confirme as mudanças Execute:

1. **git commit -m** *“texto que explica o que está sendo entregue” ou*

git commitEXEMPLO-APLICACAO-GIT-CMD – m *“texto que explica o que está sendo entregue”*

1. Dê um Git Status da Branch e veja que não existe mais nada pendente para atualizar na Master
2. Dê um **git diff** <arquivo> para ver mudanças
3. Dê um **git log** <arquivo> para ver as atualizações

10º) Atualize o Git Hub remoto com:

1. **Git push -u origin** *manut01*

NESSE MOMENTO, CONSULTE O GITHUB – A SUA BRANCH LOCAL TERÁ SIDO REGISTRADA NO GIT E VAI ESTAR HAILITADO O BOTÃO COMPARE & PUSH PARA VOCÊ PUBLICAR AS MUDANÇAS NA MASTER, SE CONCORDAR.

COMO GERALMENTE TEMOS VÁRIOS DESENVOLVEDORES TRABALHANDO EM UM PROJETO E ATUALIZANDO O REPOSITÓRIO CENTRAL, TODA VEZ QUE VOCÊ FOR USAR O SEU REPOSITÓRIO LOCAL, TERÁ QUE ATUALIZÁ-LO COM A ÚLTIMA VERSÃO DO SOFTWARE, QUE CONSTA NO GITHUB!

PARA SIMULAR ESSA SITUAÇÃO...

APÓS ATUALIZAR A MASTER NO GITHUB COM AS SUAS ALTERAÇÕES LOCAIS, ALTERE UM ARQUIVO DIRETAMENTE NO GITHUB, POR EXEMPLO O README.MD E ATUALIZE O SEU PC LOCAL PARA PODER TRABALHAR COM OS FONTES DO PROJETO MAIS ATUAIS, PUXANDO DO GITHUB A ÚLTIMA VERSÃO – SIGA OS PASSOS:

Estando no diretório do projeto no PC local, rodando o GIT CMD, execute:

1. **Git fetch** e depois **git merge** OU

**Git pull**

1. **Git status**

SE DESEJAR DELERAR A SUA BRANCH

1. **git branch -D** <nome da branch>

AO FINALIXAR, FAÇA LOGOUT NA SUA CONTA REMOTA:

1. **git config --global --unset user.email***<seu email>*

TENTE REFAZER OS PASSOS COM NOVOS ARQUIVOS.

SIMULE AS SITUAÇÕES REAIS, COMBINANDO COM UM COLEGA DA SALA, FAZEREM ALTERAÇÕES EM ARQUIVOS DIFERENTES DO REPOSITÓRIO, CADA UM COM A SUA BRANCH LOCAL, PUBLICANDO NA MASTER REMOTA.

REFLITA AS ALTERAÇÕES DO COLEGA NA SUA BRANCH LOCAL, ATUALIZE OU INCLUA OUTRO FONTE E CATALOGUE NOVAMENTE NO REPOSITÓRIO REMOTO.