**Utilizando o Git / GitHub**

**1º** Baixar e instalar o cliente git acessando o link:

https://git-scm.com/download/win

**2º** Após a instalação criar uma pasta no seu computador que será utilizada como repositório local

**3º** Clica com o botão direito do mouse e escolher a opção: **Guit Bach Here.**

**4º** Com o prompt de comando do git aberto, digite o comando: **git init.** Com isso, será criado

um arquivo oculto de nome .**git**

**5º** Adicionar a essa pasta o arquivo que será utilizado para criar uma versão junto ao seu repositório

**6º** Para verificar o status do arquivo adicionado, digite o comando: **guit** **status.** Você verá o arquivo sinalizado em **vermelho**, pois ele ainda não está pronto para ser **commitado**

**7º** Digite o comando: **git add** . Com isso, não só o arquivo que está sinalizado em **vermelho**, mas, todos serão preparados e sinalizado com a cor **verde** para ser commitado.

**8º** Digite o comando: **git commit -m "comentário curto explicando o que foi alterado".**

**9º** Será necessário autenticar com o **github**, digitando os comandos:

**git config --global user.email "seu\_email"**

**git config --global user.name "IdentificaçãoDoArquivo"**

**10º** Será necessário entrar no site do **github**, e criar um repositório. É bem simples. Após criado, salve o link de url que aponta para o seu repositório criado.

**11º** Para que seja feita a sincronia com seu repositório, digite o comando: **git remote add origin https://github.com/marcioplay/RepositorioPrincipal**

**12º** Em seguida digite o comando: **git push --set-upstream origin receiving**. Para validar a ação, siga até o seu repositório web e você irá verificar os arquivos que foram adicionados a **branch** **receiving**.

**13º** Para você ver o histórico de versões, rode o comando: **git reflog**. Será exibido todos os **commit** vinculados ao **ID**. Observar que a versão mais recente será a do topo. Aversão inicial vem com a sinalização (**initial)**

**14º** Em caso de erro na última versão, você poderá voltar para o código anterior rodando o comando: **git reset --hard IDdaVersãoAnterior.** Bem como, desfazer e retornar para a versão que foi alterada utilizando a mesma sintaxe.

**Obs.**: Essa interação entre versões, só é possível por causa do **commit**. Pois com ele, é setada as versões de forma única. Por isso a facilidade em trabalhar com versionamento.

**15º** Para verificar quantas **branchs** estão disponíveis, rode o comando: **git brach.**

**16º** Para criar uma nova branch no git, pois não é recomendável trabalhar direto master. Rode o comando: **git branch nomeDaBranch** . A palavra "receiving" quer dizer recebendo.

17º Após criada uma nova **branch**, rode novamente o comando: **git branch**. Você verá que foi adicionada uma nova **branch** de nome "**receiving**". Observar que a **branch** **master** estará em **verde** e com \*. Isso quer dizer que ela está como principal.

18º Para alterar a **branch** principal, rode o comando: **git checkout receiving**. Com isso, a **branch** principal não mais será a master.

19º Fazendo um teste no arquivo que foi commitado, e depois salvando. Serpa possível identificar, que as alterações não mais serão apontadas para a master, e sim para a branch "**receiving**".

20º Rodando o comando: **git status**, será possível identificar que o arquivo teve alteração. Porém, o arquivo master não será modificado.

21º Para ver as alterações, siga todo procedimento padrão como mencionado nos passos:

git add .

git commit -m "Comentário"

git push

git push --set-upstream origin receiving

22º Você verá que uma nova branch irá aparecer no ambiente **github web.**

23º Para executar o merge, primeiro rode o comando: **git branch**. Seguindo a sequência, mude para a branch master com o comando: **git checkou master**. Depois rodar o comando: **git pull**. Por fim, para puxar as atualizações da branch criada: "receiving", rode o comando: "**git merge receiving".**

24º Por fim, será possível ver o resultado sinalizado as alterações dos arquivos para a master.

EX:

$ git merge receiving

Updating 7f6233f..3c78041

Fast-forward

Codigos C#.cs | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)