**Criando um Repositório no GIT**

**Aula 98**

**Controle de Versão – Transformar uma pasta em um controle de versionamento.**

1º Criar a pasta do repositório

2º Criar o arquivo dentro da pasta do repositório

3º Criar um arquivo chamado “.gitignore”

OBS.: TALVEZ o Windows não permita, então o que deve ser feito é:

Criar um arquivo. Ex.: git.text

SHIFT + TAB + BTN DIREITO MOUSE na pasta

Clicar em abrir o PowerShell/Tela de Comando

Digitar: ren git.txt .gitignore

**O que significa este arquivo?**

Dentro deste arquivo contém instruções sobre o que o repositório o git vai ter que ignorar ou seja, quais arquivo não precisam ter a versão controlada.

**E como saber quais arquivos podem ser ignorados para realizar um controle de versionamento?**

1º Ir no google e digitiar: gitignore visual studio 2019 community.

2º encontrar um link que contém “git”  
3º Dentro deste link serão encontrados tudo que pode ser ignorado dentro de um projeto do visual studio.

4ª Ctrl v + Ctrl c dentro do arquivo .gitignore.

5ª Feito isto, entrar no GitBash e digitar: git init

6ª Será criado automaticamente a pasta .git  
7ª Não mexer nesta pasta.

**Aula 99**

**Identificação de Usuário no Git**

Sempre que for feita alguma alteração no repositório de versionamento, é importante identificar quem fez esta alteração.

Abrir o GitBash na pasta e digitar:

git config --global user.name “Cézar”

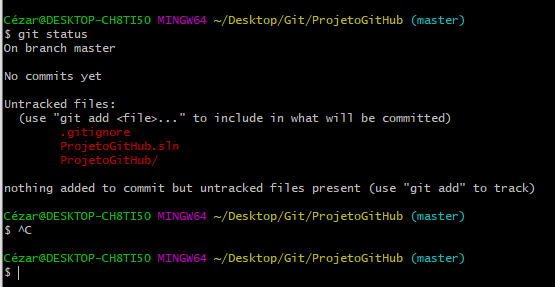
git config --global user.name “cezar88augusto@hotmail.com”

Tudo que for alterado vai ser identificado que o usuário X com o email X que alterou.

**Aula 100**

**Salvando versões**

Para saber a versão do repositório:  
Ir em GitBash e digitar: git status

Desta forma, será mostrado os arquivos que NÃO ESTÃO RASTREADOS (em vermelho)  


Visto, isto, deve-se colocar os arquivos prontos para serem “comitados” ou seja, na área de “stage”

Adicionar todos os arquivos: **git add .**

Realizar o commit**: git commit -m** “Projeto iniciado, com um olá mundo na tela”

Verificar os commits (quem fez e o email): **git log** (forma resumida: **git log –oneline**)

Fazer uma alteração no projeto, e depois verificar o status no gitbash através do cmd: **git status**

**Aula 101**

**E se eu esquecer de especificar a msg no commit?**

O GitBash abrirá um editor de texto. Para sair, apertar esc e digitar **:q!** e dar enter.

**Aula 102**

**E se eu me perder e quiser voltar o commit atual?**

Git clean -df

Git checkout -- .

**Aula 103**

**E se eu quiser desfazer o último commit?**

**Remover o último commit mantendo as alterações nos arquivos:**

git reset –soft HEAD~1

Feito isto, se der um git log, será mostrado que o arquivo sumiu, porém, se der um git status, será mostrado que há um arquivo em área de stage.  
  
**Remover o último commit inclusive as alterações nos arquivos (PERIGO):**

git reset -hard HEAD~1

**Aula 104**

**E se eu quiser somente dar uma olhada em uma versão anterior?**

Primeiro ver se tem algo pendente de commit: git status  
Depois: git checkout (+ o código do commit)

O GitBash voltará para o a versão digitada!!!  
  
E se eu quiser voltar pra versão atual?

git checkout ‘master’

**Aula 107**

**Criando um Repositório no GITHUB**

Ir no github.com/new

Adicionar novo projeto

**Associando um repositório a um repositório do GitHub**

Após o repositório ter sido criado, pegar a linha

git remote add origin <https://github.com/cezar88augusto/curso-git.git>

Lançar no gitbash do projeto que se deseja associar.

**Alterar o repositório – Associar um repositório local a um repositório remoto, porém mantendo o mesmo apelido:**

git remote set-url origin <URL do repositório remoto>

**Aula 108**

**Transferindo o repositório local para o GitHub**

git push -u origin master

**Aula 109**

**Comando de clonar o repositório remoto:**

Exemplo:

git clone https://github.com/cezar88augusto/curso-git.git

**OBS.: SE BAIXAR OS ARQUIVOS, O GITHUB NÃO PUXA O HISTÓRIO DE ALTERAÇÕES!!! Para isto, deve ser fito o git clone.**

**Aula 110**

**Atualizando o repositório:**

Dar um git log e comparar os commits com o repositório do github.com.  
  
Comando: git pull origin master