O que é controle de versionamento?

sábado, 24 de junho de 2023

20:08

* + Método usado para gerenciar o código fonte
  + Desenvolvedores podem trabalhar em diversos tipos de versões diferentes
  + É possível versões de um software em diferentes estágios

O que é um repositório?

terça-feira, 20 de junho de 2023

14:02

* + Onde ficará armazenado seu código
  + Cada projeto tem um repositório (Blog, Api, Ecommerce)
  + Criação de uma repo é o início de um projeto
  + Servidores de repo - Github e Bitbucket
  + Cada repo pode ser baixado por um desenvolvedor para fazer várias versões diferentes - Clonagem do repositório

Criando repositórios

terça-feira, 20 de junho de 2023

14:11

* git init - Criação de uma repo através da pasta que você iniciou
* A partir do git init são criados arquivos necessários para aceitar códigos git, ficam armazenados em uma pasta oculta .git
* Quando criamos uma pasta no nosso computador e utilizamos o comando git init estamos criando um repositório nessa pasta que poderá ser enviado para um servidor posteriormente.

O que é um github?

terça-feira, 20 de junho de 2023

14:20

* É um serviço para gerenciar repositórios, quando precisamos formatar nosso computador e não queremos perder nossos projetos, podemos disponibilizar no github e depois recupera-lós
* Podemos disponibilizar no github os nossos projetos para outros dev's

Observação sobre a Branch principal

Há um tempo atrás houve uma mudança no GitHub, o branch principal se chamava **master**e agora se chama **main**

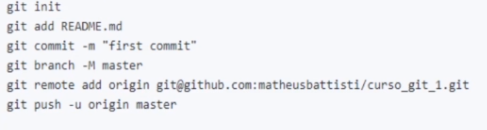
Enviando repositórios para o GH

terça-feira, 20 de junho de 2023

14:30

* Precisamos criar um repo no servidor github para que seja sincronizado com a pasta local e enviado
* São executados com poucos códigos e é feito apenas 1 vez projeto

* Já usamos o comando git init agora podemos adicionar os arquivos e enviar um commit ( mensagens ) para o repositório

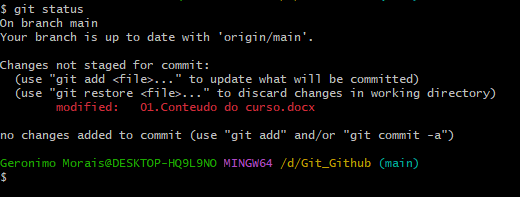


**Verificando alterações - git status**

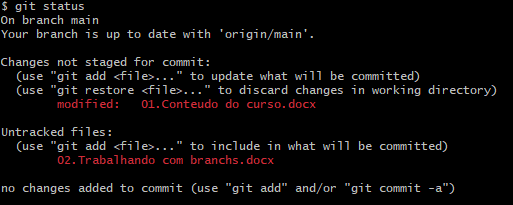
terça-feira, 20 de junho de 2023

15:03

* Deve ser utilizado o código **git status**
* verifica as modificações feitas nos arquivos
* é a diferença do que já foi enviado ao servidor ou salvo no projeto



quando usamos o **git status** mostra que o arquivo "conteudo do curso" foi modificado, existe também o untracked que é quando o arquivo foi adicionado a pasta local e ainda não foi adicionado ao repo (github)



**Adicionando arquivos ao projeto - git add**

terça-feira, 20 de junho de 2023

15:32

* + Deve-se utilizar o comando git add
  + Pode ser adicionado 1 arquivo ou vários de uma vez
  + Somente adicionando os arquivos eles estarão sendo monitorados pelo git, se não adicionados não estará no controle de versão

**Salvando alterações - git commit**

terça-feira, 20 de junho de 2023

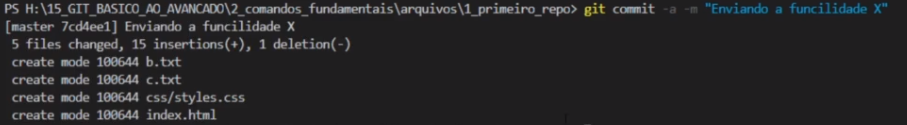
15:36

* As alterações do projeto será salvo
* Deve-se usar o comando **git commit**
* flag **-a -** envia para todos os arquivos
* Mensagem pode ser adicionada com a flag **-m**

**Salvando arquivo especifico**



**Salvando todos os arquivos com commit**



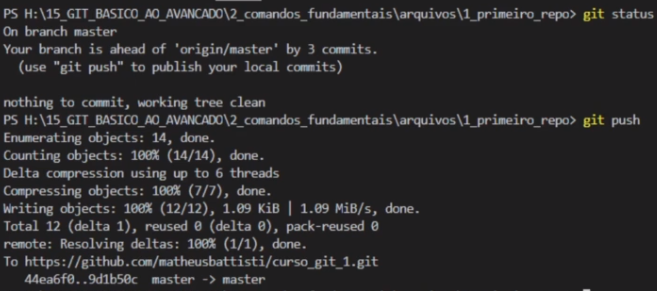
**Enviando arquivos para o repo remoto - git push**

terça-feira, 20 de junho de 2023

15:41

* Quando finalizamos o commit e todos os outros comandos, precisamos enviar ao código fonte
* Deve-se usar o **git push**
* Após isto, o servidor será atualizado baseado no repo local ( Pasta )

**Your branch is ahead of origin/master - sua branch está a frente da branch master**



* Quando finalizamos o git push o nosso repositório local está parelho com o repositório remoto

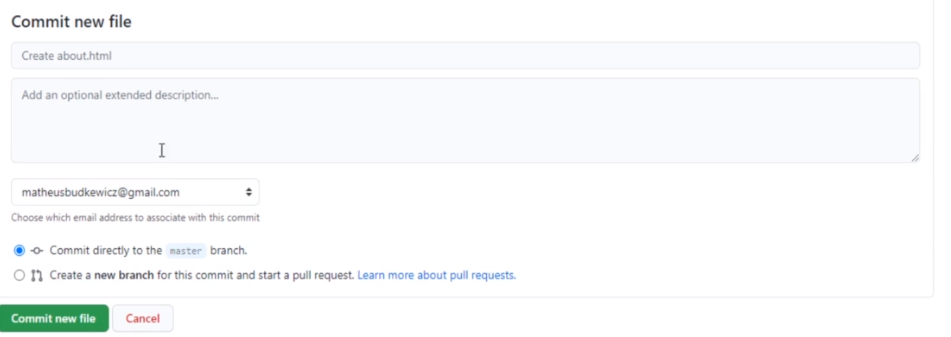
**Recebendo alterações - git pull**

terça-feira, 20 de junho de 2023

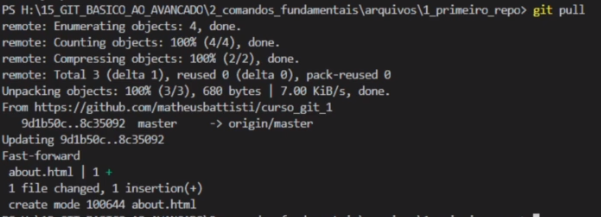
16:34

* Funciona como busca de atualizações dos códigos
* Deve-se usar o comando **git pull**
* Após o comando seram feitas buscam por atualizações, caso seja encontrada serão unidas ao seu repo local.

**Criação de um arquivo no repositório online**



**Recebendo alterações para o repositório local ( Pasta )**



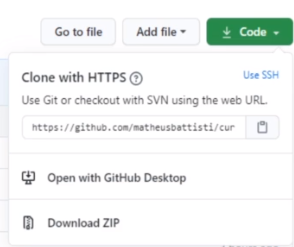
**Clonando repositório online - git clone**

terça-feira, 20 de junho de 2023

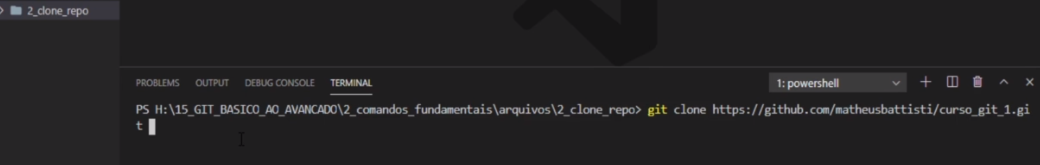
17:10

* Baixar o projeto do servidor
* Deve-se usar o **git clone**
* Pegando referências do repositório remoto
* Geralmente utilizado quando entramos em um projeto

**Copiando URL do projeto**



**Clonando repositório online**



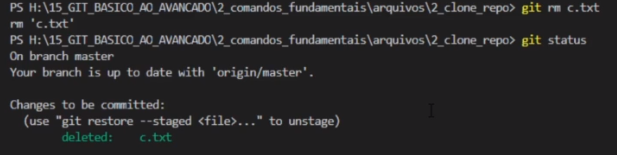
**Removendo arquivos - git rm**

terça-feira, 20 de junho de 2023

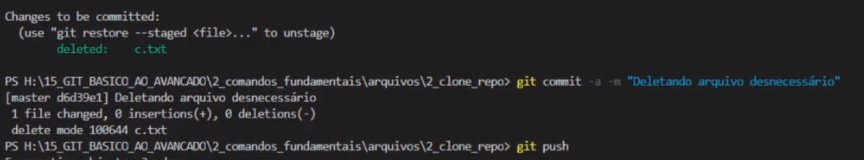
18:10

* Deve-se usar o comando git rm
* Arquivos deletados da monitoração do git
* Não terá mais atualizações pelo git após a exclusão, apenas quando for adicionado pelo comando git add

**Excluindo um arquivo**



**Enviando commit com deleção**



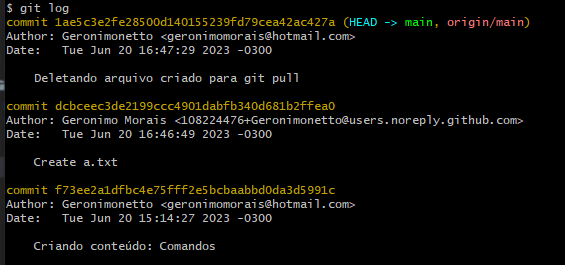
**Verificando alterações - git log**

quarta-feira, 21 de junho de 2023

18:57

* Deve-se usar o comando **git log**
* Mostra todos os commits feitos no projeto até agora

Mostrando os commits feitos - ordem decrescente



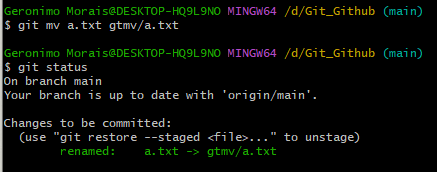
**Movendo arquivos ou renomeando - git mv**

quarta-feira, 21 de junho de 2023

19:17

* Deve-se usar o comando **git mv**
* É usado para mover arquivos para uma outra pasta
* O arquivo anterior é excluido e é criado um novo

**Movendo arquivo para uma pasta**



**OBS:** Caso o arquivo seja novo, devemos primeiro add para o monitoramento do git para depois renomear ou alterar sua pasta.

usamos o comando **git mv** nomedoarquivo.formato pastadestino/nomedoarquivo.formato

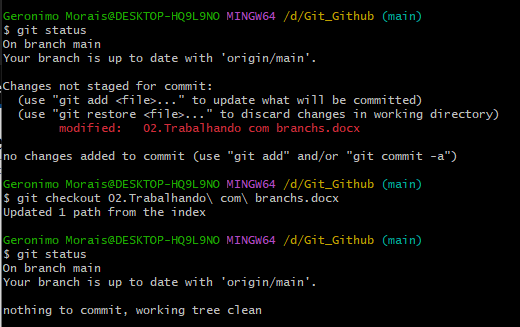
**Desfazendo alterações - git checkout**

quarta-feira, 21 de junho de 2023

19:31

* Deve-se usar o comando **git checkout**
* O arquivo volta ao seu estado original do github (**Rollback**)
* Após o uso do git checkout o arquivo sai do staging
* Caso feita uma alteração novamente entra em um staging novamente

**Cancelando modificação do arquivo para o estado original do repositório do github**



git checkout nomedoarquivo.formato

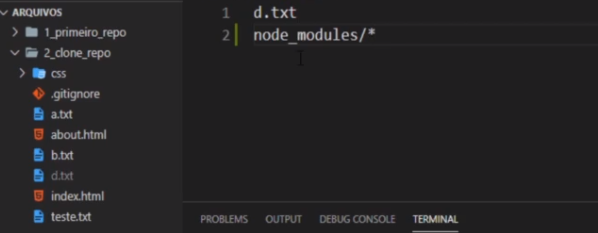
**Ignorando arquivos - .gitignore**

quinta-feira, 22 de junho de 2023

16:53

* É uma forma muito utilizada por desenvolvedores para ignorar extensão ou arquivos específicos desnecessários ou que outras pessoas não precisam saber
* Deve-se criar um arquivo na raiz do projeto com o nome **.gitignore**
* Existem arquivos gerados automaticamente que são ignorados dessa forma
* Podemos inserir arquivos que não devem entrar no versionamento ( Github )
* Primeiro deve-se adicionar no gitignore antes de adicionar ou fazer a criação automática do arquivo, ou seja caso utilize algum projeto que precise ser ignorado, coloque o nome dos arquivos ou pasta antes de iniciar o projeto.

**Ignorando arquivo específico ou pasta e tudo que estiver dentro dela**



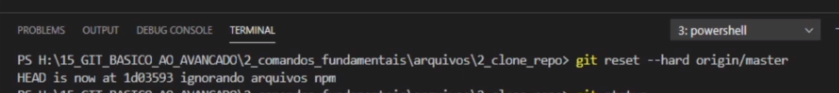
**Resetando uma branch - git reset**

quinta-feira, 22 de junho de 2023

17:11

* Reseta as mudanças feitas
* Utilizado com a flag --hard
* Todas os commit's feitos no ultimo push e também pendentes são excluidos

**Resetando a última alteração feita na branch master**



**O git reset deve ser usando antes do push**

**Ex:** Fizemos várias alterações e estão comitadas ou precisando adicionar, porém nosso repositório origin ( original - repositório remoto ) não foi alterado podemos usar o git reset --hard origin/main para excluir todos os commit's e alterações feitas nos arquivos para a forma original no repositório remoto.

Comandos

terça-feira, 20 de junho de 2023

14:19

* **git init** - Iniciando uma repo

* **git add nomearquivo** - trackeia um arquivo para ficar pronto pro commit

* **git commit -m "Mensagem do commit"** - Sinaliza uma mensagem para enviar para o repo como anotação

* **git branch -** M nomedabranch - Criação de uma branch no repositório local

* **git status -** verificar mudanças feitas nos arquivos

* **git add nomearquivo** - Adicionando arquivos para serem commitados
* **git add .** - Adicionando todos os arquivos de uma vez

* **git rm -** removendo arquivos

* **git log -** Verifica todos os commit's feitos até o momento detalhadamente
* **git log --oneline** - Mostra todos os commits em uma linha só

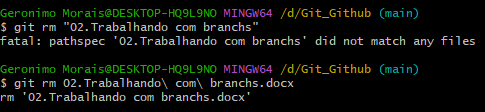
* **git revert HEAD** - Reverte o último commit feito

**Casos específicos**

terça-feira, 20 de junho de 2023

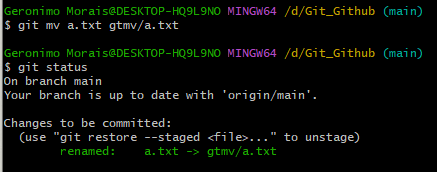
18:20

**Caso o arquivo contenha espaços no nome como excluir?**



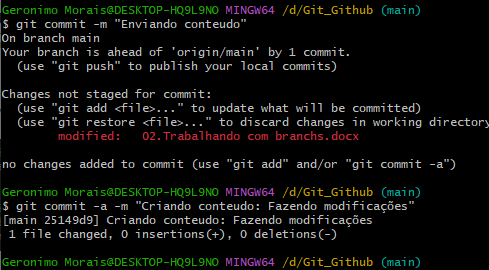
deve-se usar \ + espaço para representar o espaço

**Movendo um arquivo**



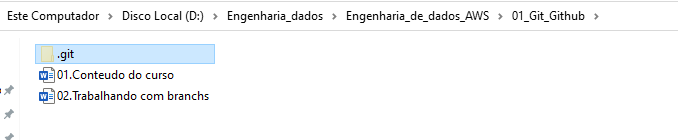
**OBS:** Caso o arquivo seja novo, devemos primeiro add para o monitoramento do git para depois renomear ou alterar sua pasta.

**Modificando arquivos**

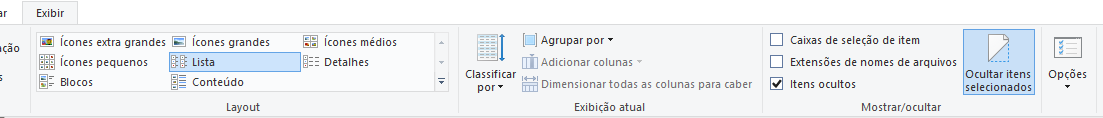


Podemos observar no exemplo a seguir, quando modificamos um arquivo já existente, não precisamos mais usar o git add, podemos apenas usar o git commit -a -m "Mensagem" para poder enviar o conteúdo para o repositório.

**Pasta inacessível dentro do git**



Essa pasta está dentro de uma pasta principal que contém o arquivo .git de início de um repositório porém ela também tem um .git dentro, a pasta no github ficará inacessível, logo devemos apagar essa pasta .git da pasta que encontra-se inacessível.



Marque a opção itens ocultos > Excluir pasta .git