



Versionamento Git & Github

Desenvolvimento Web

VIDA DE PROGRAMADOR

.COM.BR

/" HISTÓRIA REAL
ENVIADA POR
LÉO HADDAD "/



#330

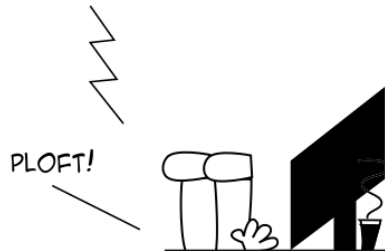
CADÊ AS CORREÇÕES DO
SISTEMAS QUE DEVERIAM ESTAR
NO AR HOJE?



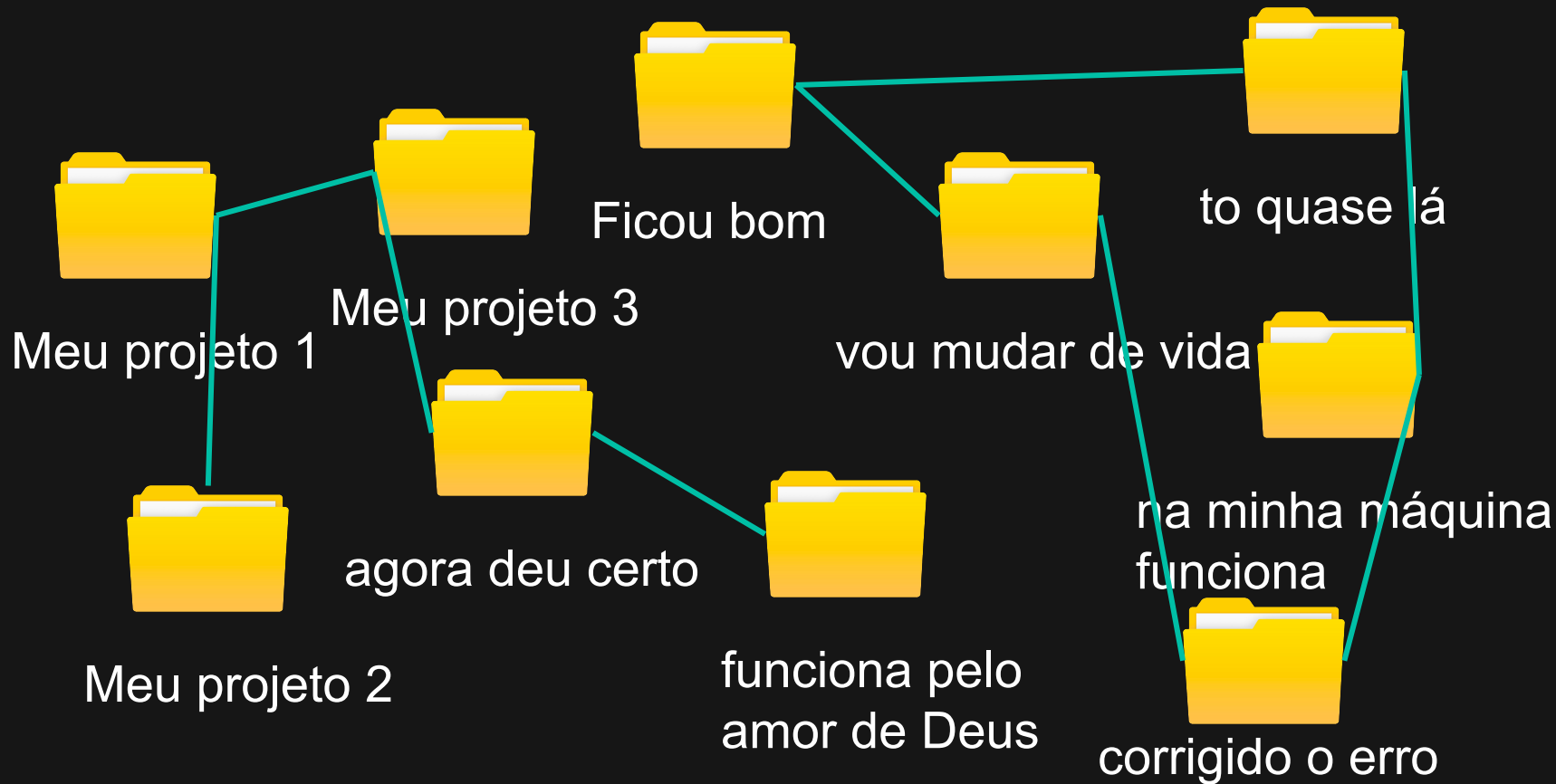
ESTOU TENDO PROBLEMAS DE
SUBIR AS ATUALIZAÇÕES PARA O
REPOSITÓRIO. ASSIM QUE
RESOLVER, COLOCO NO AR
E TE AVISO.



TÁ, OK. COLOCA
O SUPOSITÓRIO AÍ E
DEPOIS NOS FALAMOS...



Eu preciso mesmo disso ?



Acho que sim



Sistemas de versionamento



git



Sistemas de versionamento



O que é o git ?

É um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo. O Git foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do kernel Linux, mas foi adotado por muitos outros projetos.



WIKIPÉDIA
A enciclopédia livre

- Página principal
- Conteúdo destacado
- Eventos atuais
- Esplanada
- Página aleatória
- Portais
- Informar um erro
- Loja da Wikipédia

Colaboração

- Boas-vindas
- Ajuda
- Página de testes
- Portal comunitário
- Mudanças recentes
- Manutenção
- Criar página
- Páginas novas
- Contato
- Donativos

Ferramentas

- Páginas afluentes
- Alterações relacionadas
- Carregar ficheiro
- Páginas especiais
- Hiperligação

Não autenticado Discussão Contribuições Criar uma conta Entrar

Artigo Discussão Ler Editar Editar código-fonte Ver histórico Pesquisar na Wikipédia

 Junte-se à Campanha Páginas da Wikipédia que Precisam de Fotos, ajude a ilustrar os artigos da Wikipédia, e ganhe um prêmio!

[ocultar]

Git

Origem: Wikipédia, a enciclopédia livre.

Git pronuncia-se /git/ (ou pronuncia-se /djít/ em inglês britânico) é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo. O Git foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do kernel Linux, mas foi adotado por muitos outros projetos.

Cada diretório de trabalho do Git é um repositório com um histórico completo e habilidade total de acompanhamento das revisões, não dependente de acesso a uma rede ou a um servidor central. O Git também facilita a reprodutibilidade científica em uma ampla gama de disciplinas, da ecologia à bioinformática, arqueologia à zoologia.^[1]

O Git é um software livre, distribuído sob os termos da versão 2 da GNU General Public License. Sua

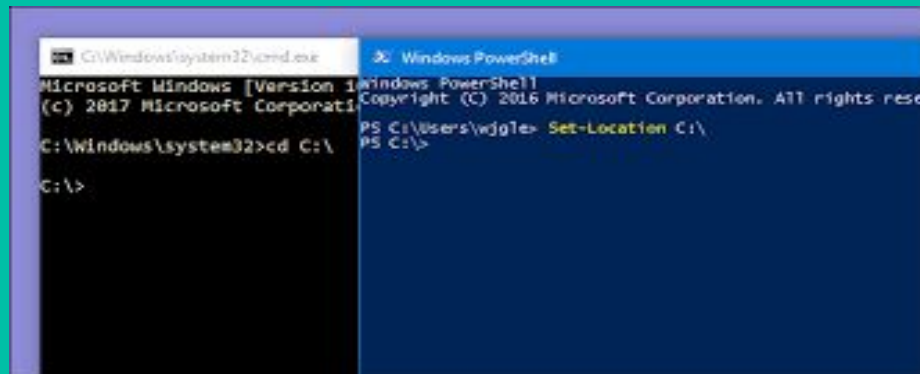
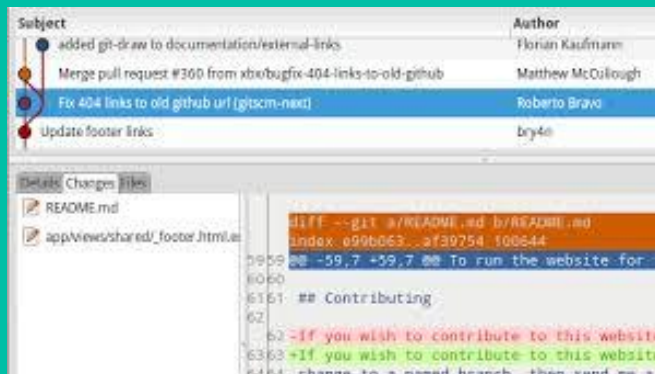
Git	
	git
Desenvolvedor	Linus Torvalds, Junio Hamano
Plataforma	Multiplataforma
Lançamento	7 de abril de 2005 (15 anos)
Versão estável	2.22.0 (7 de junho de 2019; há 14 meses)
Mercado-alvo	Versionamento de Software
Linguagem	C, Shell, Perl
Sistema	POSIX

Instalando o Git

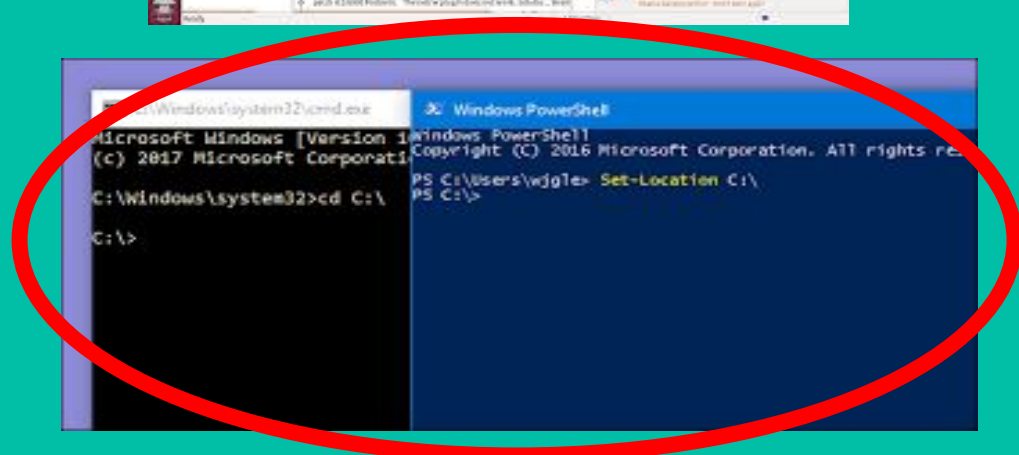
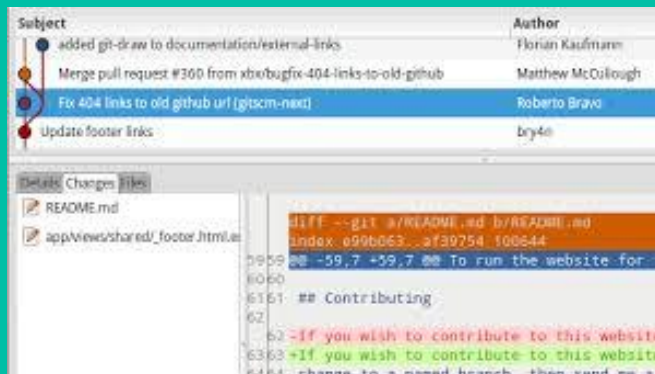
Link documentação Oficial

<https://git-scm.com/book/pt-br/v2/Come%C3%A7ando-Instalando-o-Git>

Como usar o git ?



Como usar o git ?



Linha de comando

Windows : powershell ou CMD ou WSL
Linux / MacOS : Bash , ZSH

O básico de Terminal

```
// Comando ( dir ) ou ( ls )
// Irá exibir o conteudo da pasta (diretorio)

C:\Users\FredericoBezerra\aula-4> dir //ou ls
Mode                LastWriteTime         Length Name
----                -
d-----          7/10/2020   1:45 AM                index.html
d-----          4/13/2020   6:31 PM                css

// Comando ( cd ) (pasta (diretorio) para abrir)
//Ira navegar para o diretorio selecionado
C:\Users\FredericoBezerra\aula-4> cd css

// note que agora vc esta dentro da pasta css no terminal
C:\Users\FredericoBezerra\aula-4\css>

// cd .. (cd + espaço + ..)
// comando para voltar para a pasta superior
C:\Users\FredericoBezerra\aula-4\css> cd ..
C:\Users\FredericoBezerra\aula-4>

// code . (abre o diretorio)
// abrir o vscode na pasta (diretorio) corrente
C:\Users\FredericoBezerra\aula-4> code .
```

Primeira configuração do git

A primeira coisa que devemos fazer quando após instalarmos o git é realizar a configuração global de **identidade** : “Quem está fazendo as alterações”.

Existem outras configurações porém por agora vamos nos atentar a estas **user.name** e **user.email**. Se atente ao espaços no comando para não haver erros.

1- Abri o cmd/powershell/terminal

2- digitar : git config --list

3- Adicionar configurações

4a - *git config --global user.name "fredd.devs"*

4b - *git config --global user.email fredd@devs.com*

***Dica : utilize seu email e username do github...*

Iniciando git em um projeto

Para inicializar o versionamento com o git em um local específico precisamos navegar até a pasta (diretório) do nosso projeto e na raiz do projeto devemos executar o comando ***git init*** com isso já podemos iniciar nosso trabalho com o git

dentro da pasta do seu projeto digite o comando :

```
C:\Users\FredericoBezerra\aula-4> git init
```

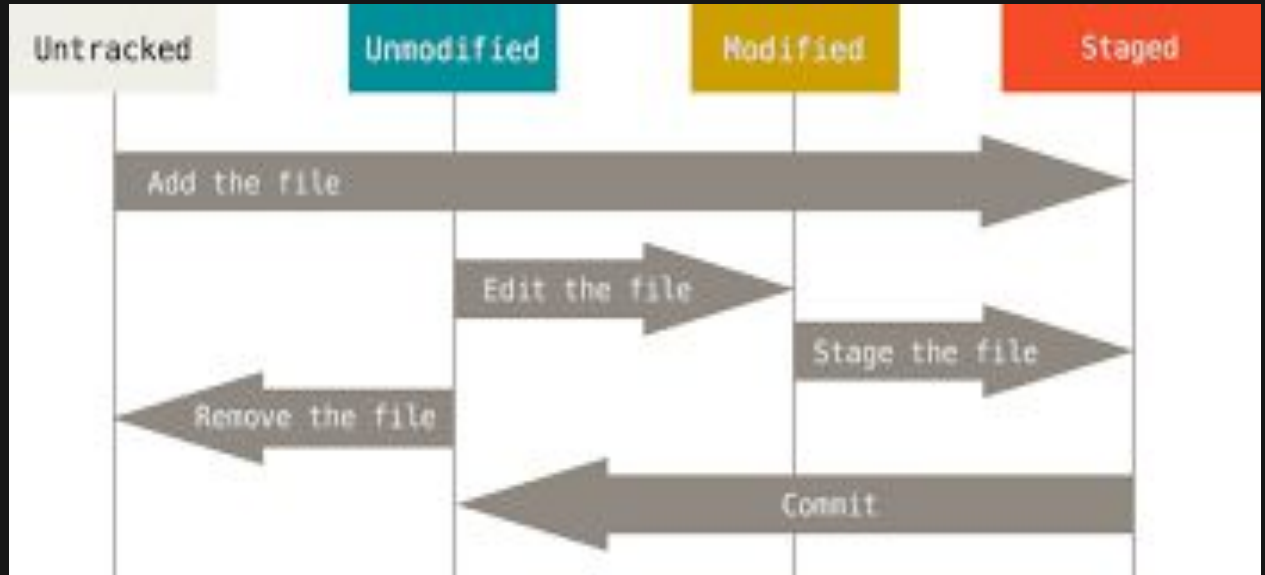
após o comando irá aparecer a seguinte mensagem :

```
Initialized empty Git repository in [ Pasta... ]
```


Trabalhando com Git

Fluxo de trabalho local com git.

- git status
- git diff
- git add
- git reset
- git commit -m



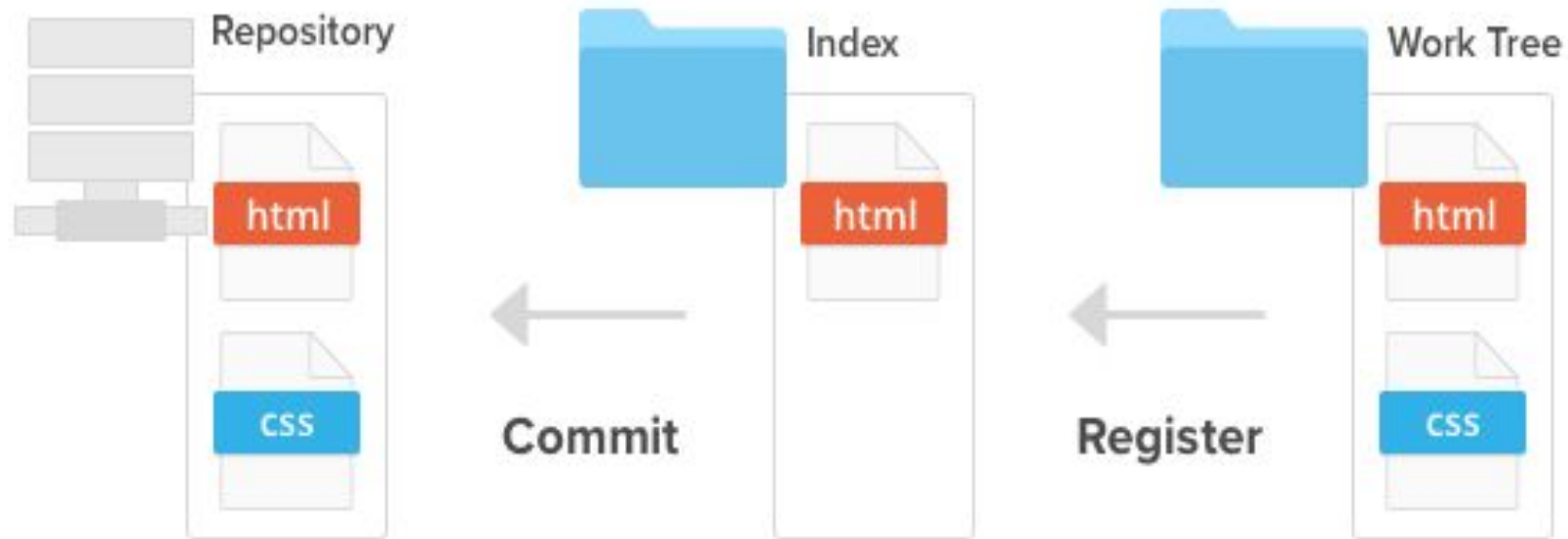
Repositório - árvore de trabalho - index

Repositório : ou **repo** é o container onde ficam rastreados as modificações do projeto. armazena todos os commits e suas modificações.

árvore de trabalho : ou **working directory** , arquivos que você está trabalhando atualmente.

index : ou **staging area** , onde os commit são preparados. O index compara os arquivos na árvore de trabalho dos arquivos do repo. Quando você faz uma mudança na árvore de trabalho o index marca esse arquivo como modificado antes de ser realizado o commit.

Repositório - árvore de trabalho - index



Repositório - árvore de trabalho - index

De Acordo com a imagem anterior o fluxo básico ocorre assim :

1 - **Modificamos** arquivos na árvore de trabalho

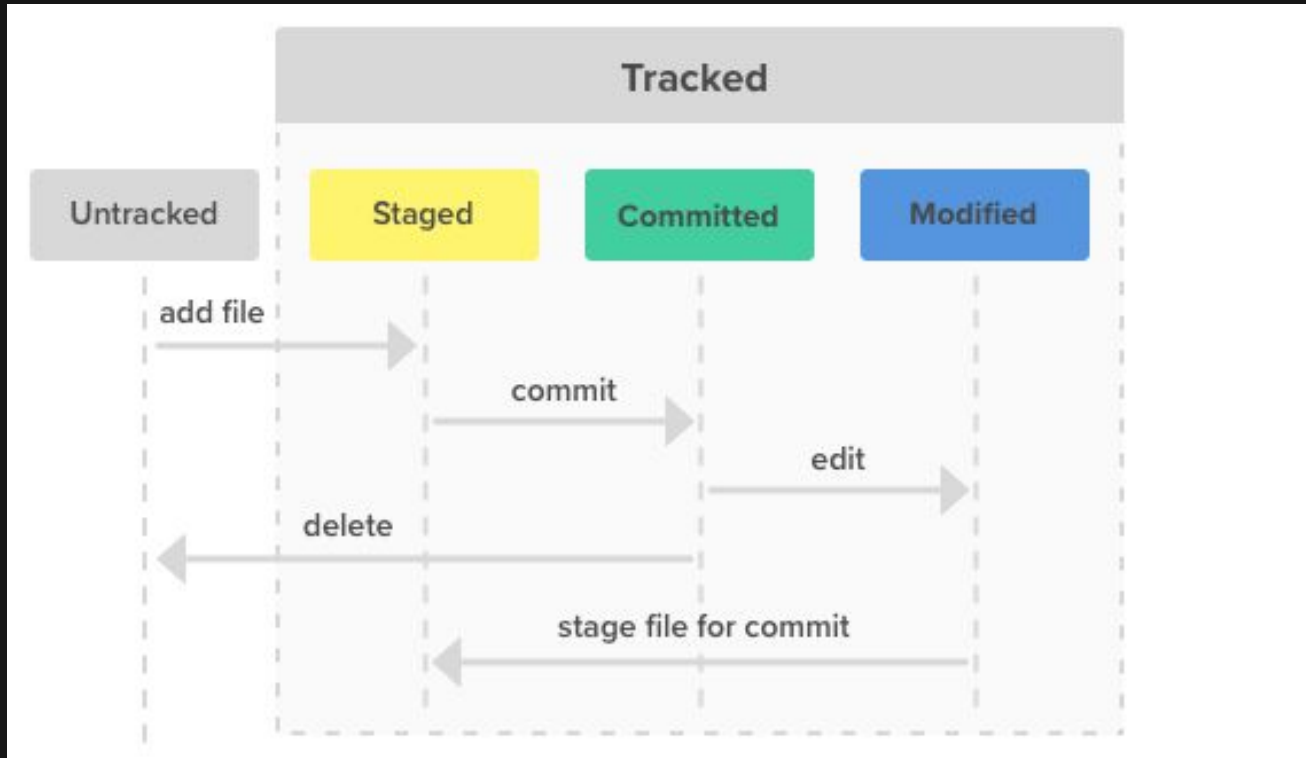
2 - **adicionamos** as mudanças (arquivos) que queremos incluir no próximo commit

3 - Realizamos o **commit** das mudanças. o Commit leva os arquivos modificados do index (staging area) e armazenam uma “Fotografia” no repositório com as modificações realizadas.

Nesse fluxo básico os arquivos podem ter os seguintes estados :

- **Modified** (Modificado)
- **Staged** (adicionado a fila de commit)
- **Committed** (Comitado hAhAhah :))

Repositório - árvore de trabalho - index



Mão na massa!
Hands On !



Mão na massa git local

1 - Criar uma pasta aula-git

2 - Criar 2 arquivos index.html ,
style.css

3 - Inicializar o repositório (Slide 15)

4 - Adicionar os arquivos da árvore de
trabalho (staging area)

5 - Realizar o commit da staging area
para o repositório

6 - Verificar o status.ção

7 - Adicionar estrutura ao arquivo
index.html

8 - Verificar o status

9 - Verificar as diferenças

10 - adicionar a modificação para a
staging area

11- realizar o commit da modificação

12- verificar o status atual

13 - Verificar o log de alteração

Mão na massa git local

1 - Criar uma pasta aula-git

2 - Criar 2 arquivos index.html , style.css

3 - Inicializar o repositório (Slide 15)

4 - Adicionar os arquivos da árvore de trabalho (staging area)

git add .

5 - Realizar o commit da staging area para o repositório

git commit -m "descrição do commit"

6 - Verificar o status.ção

git status

Mão na massa git local

7 - Adicionar estrutura ao arquivo index.html	
8 - Verificar o status	git status
9 - Verificar as diferenças	git diff
10 - adicionar a modificação para a staging area	git add index.html
11- realizar o commit da modificação	git commit -m "descrição do commit"
12- verificar o status atual	git status
13 - Verificar o log de alteração	git log

Mão na massa git local

Inicializar um repositório (.git)	git init
Verificar status dos arquivos na área de trabalho	git status
Verificar diferenças entre os arquivos modificados	git diff
Adicionar arquivo para staging area (fila para o commit)	git add index.html
Realizar o commit , envia arquivos da staging area para o repositório.	git commit -m "descrição do commit"
Exibi o log de alterações "commits"	git log

Trabalhando em equipe com Git



Trabalhando em equipe com git

- git clone
- git fetch
- git branch
- git checkout
- git pull

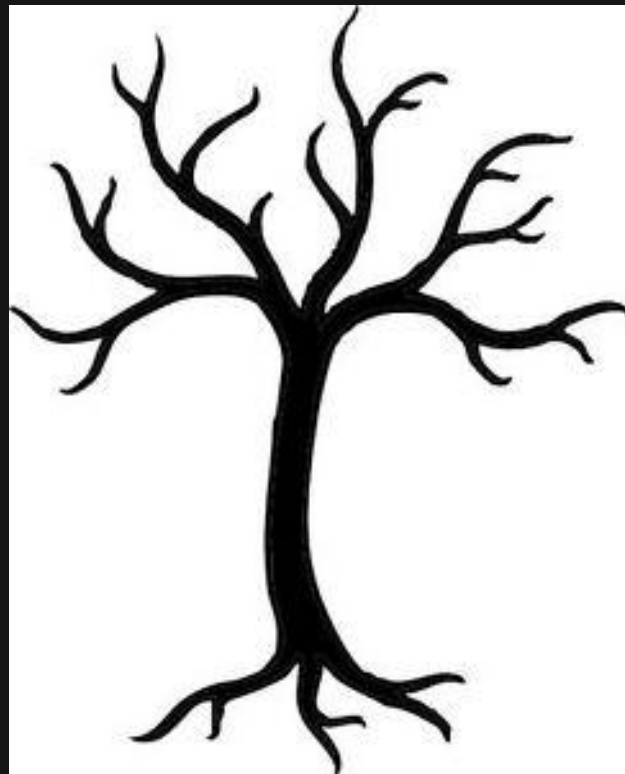
Repositório remotos

- github

Trabalhando em equipe com git

O que é uma branch ?

Uma ramificação no git é um ponteiro para as alterações feitas nos arquivos do projeto. É útil em situações nas quais você deseja adicionar um novo recurso ou corrigir um erro, gerando uma nova ramificação garantindo que o código instável não seja mesclado nos arquivos do projeto principal. Depois de concluir a atualização dos códigos da ramificação, você pode mesclar a ramificação com a principal, geralmente chamada de master.



Trabalhando em equipe com git

O que é um repositório remoto?

É um repositório que não está presente na nossa máquina local , está em um servidor na rede ou um servidor na internet.

No nosso caso utilizaremos o github como repositório remoto de nossos projetos.

Para utilizar o github você precisa fazer o cadastro com email e senha.



Trabalhando em equipe com git

git remote add [local do repositório remoto]

adicionar a URL do repositório remote que iremos utilizar para o git fetch ,git pull e git push.

Faz o download

git remote add [local do repositório]

Trabalhando em equipe com git

git clone realiza a cópia de um repositório remoto a partir do seu endereço. com ele traz todas as informações de commits pré existentes.

Realiza o clone de um repositório remoto

git clone [endereço do repositório]

Trabalhando em equipe com git

git fetch é o primeiro comando usado para trabalho em equipe. com ele realizamos o download de todos os dados remotos de um repositório para iniciarmos o desenvolvimento.

Baixa o conteúdo do repo remoto

git fetch

Trabalhando em equipe com git

git branch -a lista todas as branches de um repositório.

com esse comando você poderá verificar as branches existentes.

Lista as branches de um repo

git branch -a

*master

feature/footer

Trabalhando em equipe com git

git checkout com esse comando navegamos entre as branches do nosso repositório.

também conseguimos criar uma branch no momento do checkout, com isso esse comando copia os dados da branch atual para criação de uma nova.

Navega entre as branches

git checkout [nome da branch]

Navega e cria uma nova branch

git checkout -b [nova branch]

Trabalhando em equipe com git

git pull faz o download das atualizações da branch corrente e também tenta realizar o merge das modificações locais com a do servidor remoto

Faz o download

git pull

faz o download forçando o que está no servidor

git pull --rebase

Trabalhando em equipe com git

git push Faz o envio de nossas modificações no repositório local para o repositório remoto.

Realiza o push de nossas modificações para a branch master.

git push origin [nome da branch master]

Realiza o push de nossas modificações para a branch master quando o repositório remoto está “setado” para a branch master ou branch corrente.

git push

Trabalhando em equipe com git

git merge [nome da branch]

Realiza o merge da branch corrente com a branch selecionada.

Realiza o merge da branch local com a branch selecionada

git merge [nome da branch]



Mão na massa!
Hands On !



TRABALHO DE CASA

- CRIE SUA CONTA NO GITHUB *
- FAÇA TODO O PROCESSO DE CONFIGURAÇÃO E INICIALIZAÇÃO DO GIT NA SUA ÁRVORE DE TRABALHO (IN LOCAL)
- CRIE UM REPOSITÓRIO NA SUA CONTA
- SUBA SEUS EXERCÍCIOS DA AULA ANTERIOR NESSE REPOSITÓRIO (PUBLICO)
- DISPONIBILIZE O LINK;