**Controle de Versão**

Gerenciamento do ciclo de evolução do código fonte

Ferramentas para controle de versão

* Git
* Subverion / SVN
* Mercurial
* CVS

Conhecendo o GIT

O git é um sistema de controle de versão distribuído e um sistema de gerenciamento de código fonte, com ênfase em velocidade. O Git foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do Kernel Linux, mas foi adotado por muitos outros projetos

Lista de Comandos GIT

* git init
* git clone
* git add . ou git add <<nome\_arquivo>>
* git commit –m ‘mensagem de commit’
* git push
* git pull
* git merge
* git status
* git log
* git branch <<nome da branch>>

Para criar um arquivo vazio pelo prompt de comando pode ser utilizado

Type nul > nomedoarquivo.extenção

visite: <https://git-scm.com/docs>

Quando estou no github e vejo um projeto legal, e quero mexer nele eu posso copiar o link e dar o comando na pasta do git localmente

git clone <https://github.com/Edsonjunior91/projeto-curso-git.git>

adicionar um repositório remoto

git remote add origin https://github.com/Edsonjunior91/projeto-curso-git.git

remover um repositório remoto

git remote rm origin

para enviar ao repositório remoto

git push –u origin máster

toda vez que fizer alteração no arquivo é necessário fazer

git add index.html

git commit –m ‘“Modificação executada”’ no Windows é necessário as aspas duplas pra definir que é caracter

e por fim para mandar o arquivo alterado para github faz

git push

pode ser dado git status para verificar como esta o arquivo no repositório local

Trabalhando com Branching

Branches permitem que diferentes versões do mesmo conteúdo se desenvolvam ao mesmo tempo, porém de forma independente, enquanto você recombina periodicamente as contribuições de diferentes branches em um processo chamado de “fusão” merging.

git branch <<nome branch>

ou

git checkout –b <<nome branch>>

O Comando git branch <<nome branch>> cria uma nova branch no repositório local, mas mantém a referência do origin para branch default no caso a máster.

Já o comando git checkout –b <<nome branch>> cria uma nova branch e já troca a referencia do origin para a nova branch criada

git branch

o comando git branch irá nos mostrar quais branches temos criada em nosso repositório

git checkout <<nome branch>> serve pra trocar de branch

Deletar uma branch

git branch -D <<nome branch>

git push origin <<nome branch>>

o comando acima irá enviar a nossa branch do repositório local para o repositório remoto

git merge <<nome branch>>

o comando git merge <<nome branch>> é responsável por mesclar alterações de uma branch, a uma branch ativa, por exemplo a branch master