

CURSO BÁSICO GIT E GITHUB
PROFESSOR: ELDO JÚNIOR
ORIENTADOR: MARCUS

INTRODUÇÃO

Em algum momento da sua vida você teve que lidar com cópias e cópias de um mesmo projeto. Organizar um projeto de modo que facilite o seu manuseio é um desafio que muitas organizações lidam diariamente. As técnicas mudaram com o tempo, hoje temos sistemas que possibilitam um melhor controle das versões de um projeto. Temos exemplos como Subversion, Mercurial... porém vamos falar de um que tem sobressaído nas comunidades de desenvolvedores, empresas em geral: **o Git!**

o Git nada mais é que um **Sistema Controle de Versão** que possibilita gerenciar diferentes versões de um documento/projeto

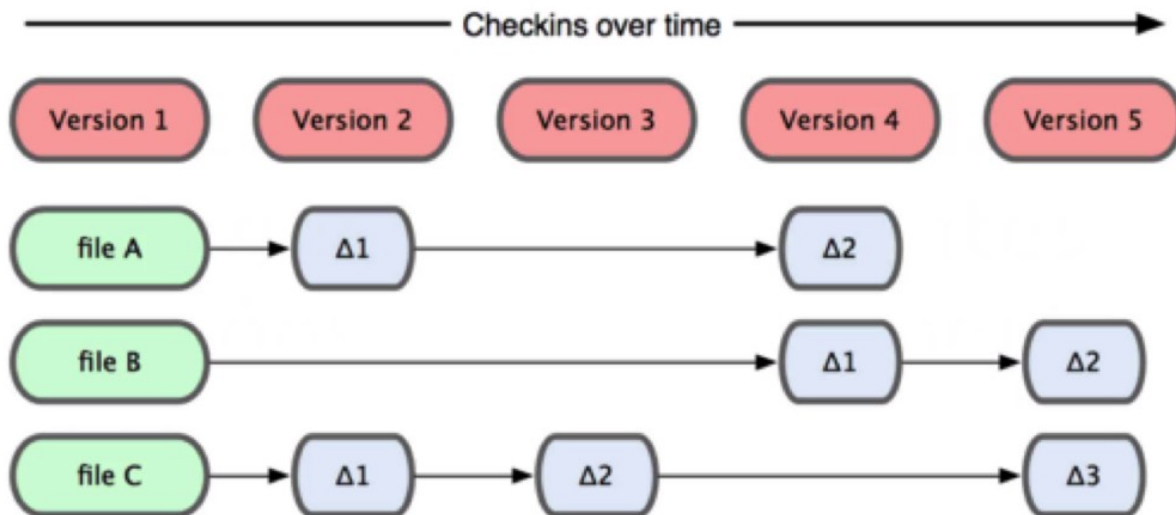
Versão: conjunto de modificações dos arquivos de um projeto.

o Git possui uma particularidade em relação aos demais **SCV's**, ele trabalha com "snapshots", ou seja um "link" dos arquivos não modificados o que facilita aos seus usuários avançar ou voltar versões. O processo é mais custoso em outros **SCV's**, pois são salvas somente as diferenças dos arquivos. Veja as imagens abaixo:



*Os pontilhados são os snapshot's.

Outros Sistemas



HISTÓRIA Do Git

A empresa BitKeeper fornecia gratuitamente um **SCV** para o projeto do kernel do linux mantido pela Linux Foundation. Em dado momento, a BitKeeper resolveu cortar essa benesse e gerou revolta de Linus Torvalds, responsável pela Linux. Linus roupeu com a BitKeeper e resolver desenvolver seu próprio **SCV** o qual o nomeou Git.

O novo **SVC**, Git, proporcionou a equipe da Linux Foundation as seguintes melhorias

- Velocidade
- Design simples
- Suporte robusto a desenvolvimento não linear
- Totalmente distribuído
- Capaz de lidar eficientemente com grandes projetos como o kernel do linux

GITHUB

Serviço de web compartilhado para projetos que utilizam o Git para versionamento.

GIT não é GITHUB

O Git trabalha no seu **repositório local** e somente envia seus arquivos para o **repositório remoto** que está na sua conta do Github.

Repositório local: A grosso modo é a pasta do seu projeto que contém o “.git” como veremos mais a frente.

Repositório remoto: Pasta contida em no Github que armazena o projeto enviado do repositório local.

CONFIGURANDO o Git

Após instalar o Git em seu computador, (consulte as instruções no site oficial <https://git-scm.com/downloads>) é necessário fazer as configurações globais. Global pois se aplica a qualquer repositório local do Git que você for manipular. É necessário informar seu nome de usuário, e-mail e o editor de texto padrão.

```
git config --global user.name "Eldo Jr." - Informa o nome de usuário  
git config --global user.email "eldojr90@email.com" - Informa o e-mail do usuário  
git config --global core.editor gedit - Informa o editor de texto padrão
```

INICIALIZANDO UM REPOSITÓRIO LOCAL

Após configurado o Git, para inicializar o repositório local, abra o prompt de comando(Windows) ou terminal(Linux ou Mac OS) e crie uma pasta. Entre nessa pasta e digite:

git init

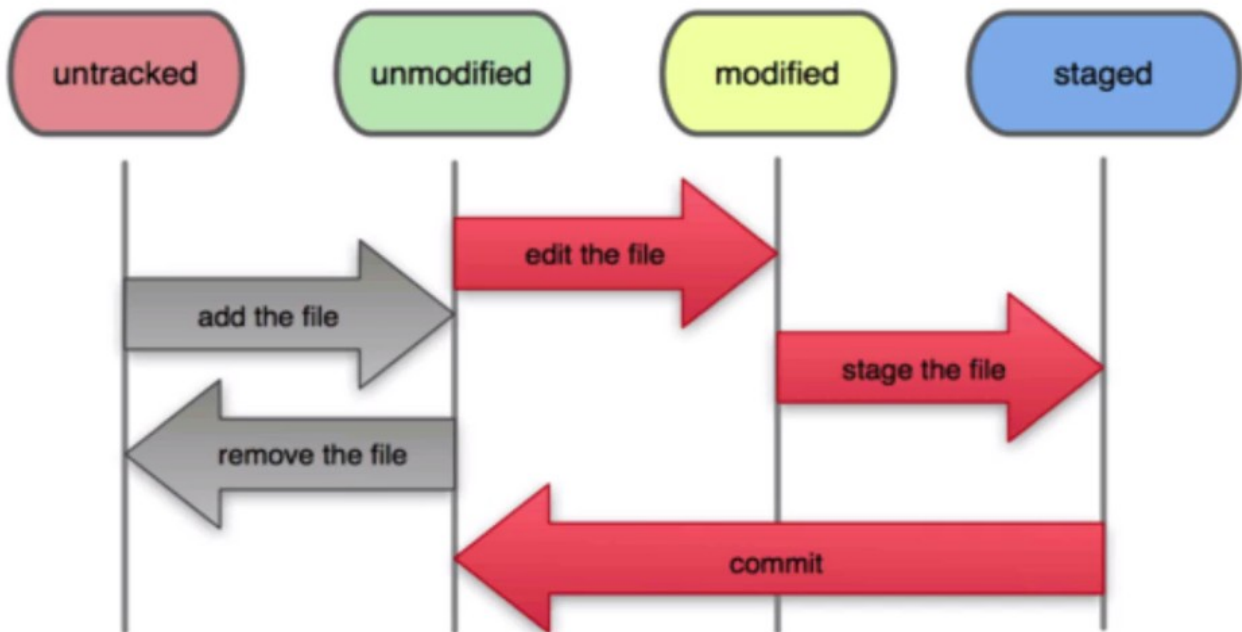
Após esse comando, exiba o conteúdo da pasta

(Comandos "dir" [Windows] ou "ls -lh" [Linux ou Mac OS])

Observe que foi criado a pasta ".git". Nela contém dados referentes ao manuseio do seu repositório, utilizados internamente pelo Git. Portanto, removê-la implica em perder o controle das versões e demais informações do repositório em questão.

ESTADOS DO ARQUIVO

File Status Lifecycle



untracked – Não marcado

O Git não monitora esse arquivo pois ele acabou de ser adicionado ou criado na pasta do projeto.

unmodified – Não modificado

O Git monitora esse arquivo porém ele não foi modificado. (Não precisa ser adicionado)

modified - Modificado

O Git monitora, porém esse arquivo não está apto para sofrer push. Está em um estado similar a "untracked" pois é necessário adicioná-lo (git add) para possibilitar o commit das alterações.

staged – Estado do arquivo ao ser adicionado

Caso o arquivo seja commitado ele passa a ser "unmodified".

PRÁTICAS:

Dando prosseguimento a prática de inicialização do repositório

Crie o arquivo index.js na pasta do projeto

Insira a linha : `console.log("exemplo1");`

Digite o comando: `git status`

Observe o estado do arquivo "index.js"

Adicione-o para o estado "staged": `git add index.js`

Verifique novamente o estado do arquivo

VISUALIZANDO LOGS

Log é um registro de uma atividade realizada por um software. Teste em seu terminal os seguintes comandos:

`git log` - informa sobre todos commits

`git log --author="Eldo Jr"` - filtra os commits pelo autor

`git shortlog` : quais foram os autores, quantos commits fizeram e quais foram

`git shorlog -sn` : quantidade de commits e quem commitou

`git log --graph`: mostra de forma gráfica os merges e rebases

`git show <hash>` : informa os dados do commit referente a hash especificada

ex: `git show 36947a9efdb9b93a617b33ca962b3fc55543ce5b`

A hash é uma chave encriptada para identificar um log.

VERIFICANDO ALTERAÇÕES DO ARQUIVO

Possibilidade de visualizar as mudanças antes que o arquivo passe para o estado "staged".

git diff - mostra o que foi modificado em um arquivo

PRÁTICA:

Altere o arquivo index.js

Dê o comando **git diff** e verifique as alterações realizadas no arquivo e outras informações

Exercícios de Fixação

1. Conceitue Git.
2. Qual a finalidade do Git?
3. Qual o diferencial do Git?
4. Qual a diferença entre repositório local e remoto?
5. Conceitue Github.
6. Informe:
 - a) Comando para inicializar repositório local:
 - b) Visualizar os estados dos arquivos do repositório local:
 - c) Colocar um arquivo no estado "staged":
 - d) Vincular ao repositório local um repositório remoto:
 - e) Comando para fazer upload dos arquivos commitados.
 - f) Comando para verificar o histórico de commits realizados:
 - g) Comando para verificar as alterações dos arquivos.