Sistema de Controle de Versão

- Controlam as versões de um arquivo ao longo do tempo:
- Registram históricos de atualizações de um arquivo
- Gerencia quais foram as alterações, a data, autor, etc.
- Organização, controle e segurança

Tipos de controle de versão:

Dentre os sistemas de controle de versão (VCS), temos:

- VSC Centralizado (CVCS)
- VCS Distribuidos (DVCS)

VCS Distribuidos (DVCS)

- Clona o repositório completo, incluindo o histórico de versão
- Cada clone e como um backup
- Possibilita um fluxo de trabalho mais flexível
- Possibilidade de trabalhar fora da rede (WEB)

O que é o GIT

E um sistema de versão distribuído

- Gratuito e Open Soucer
- Ramificação (branching) e fusões (merging) eficiente
- Leve e rápido

Breve histórico do GIT

- 2002 projeto do karnel do LINUX, que e open soucer e começou a utilizar o **Bitkeeper,** um **DVCS proprietário;**
- 2005 Após com a comunidade, o Bitkeeper rescinde a licença gratuita. O que leva a **Linus Torvalds,** criador do **Linux,** e sua equipe desenvolverem sua própria ferramenta **GIT.**

O que é o Git Bash?

O Git Bash é o aplicativo para ambientes do Microsoft Windows que oferece a camada de emulação para a experiência de linha de comando Git. Bash é acrônico para "Bourne Again Shell". Shells são aplicativos terminais usados como interface em sistemas operacionais por meio de comandos gravados. O Shell é o padrão popular no Linux e no macOS. O Git Bash é o pacote que instala o Bash, alguns utilitários de bash comuns e o Git nos sistemas operacionais Windows.

O Git Bash é empacotado com comandos adicionais que podem ser encontrados no diretório /usr/bin da emulação. O Git Bash oferece uma experiência robusta de shell no Windows. Ele vem com os comandos de shell a seguir, que estão fora do escopo deste documento: Ssh, scp, cat, find.

Além do conjunto mencionado de comandos do Bash, o Git Bash inclui o conjunto completo dos principais comandos do Git discutidos nesse site. Saiba mais nas páginas de documentação correspondentes do git clone, git commit, git checkout, git push e muito mais.

Comandos do terminal:

- **cd:** muda de diretório exemplo cd d:\\pasta\subpasta
- **mkdir:** cria diretório mkdir nome do diretorio
- **touch:** cria arquivo exemplo nome do touch arquivo.extensão
-

Comando do GIT

- **git clone:** Clona um repositório GIT existente para um novo diretório local
- **git commit:** grava alteração feita no repositório

- **git pull:** puxa as alterações do repositório remoto para a pasta local (busca e mescla)
- **git push:** empurra as alterações do repositório local para o repositório remoto
- **git checkout:** Visualiza commits antigos
- **git fetch:** baixa commits, arquivos e referencias de um repositório remoto, para o repositório local.
- **git merge:** Combina varias sequencias de commits em um histórico unificado
- **gitignore:** criar arquivo para ignorar pasta na hora de sincronizar com o github exemplo echo none da pasta \> .gitignore
- **gitkeep:** Cria um arquivo gitkeep pra que se possa reconhecer um diretório ou arquivo vazio exemplo: touch nome da pasta /.gitkeep
- **rm -rf .git:** Remove recursivamente uma repositório GIT
- **git restore:** volta arquivo modificado ao estado que se encontrava no ultimo commit.
- **git amend:** usado para corrigi descrição de um commit, exemplo git commit –amend -m "Descrição Atualizada"
- **git reset:** desfaz o ultimo commit exemplos: **git reset –soft** + endereço do commit (pega os arquivos dos commit posteriores e adiciona a área de preparação apagando totalmente o commit); **git reset -mixed**

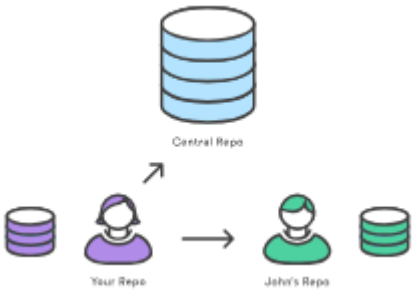
GIT Remote

O SVN usa um único repositório centralizado para servir como hub de comunicação para desenvolvedores e a colaboração ocorre passando os conjuntos de alterações entre as cópias ativas dos desenvolvedores e o repositório central. Isso é diferente do modelo de colaboração distribuída do Git, que dá a cada desenvolvedor sua própria cópia do repositório, completa com seu próprio histórico local e estrutura de ramificação. Geralmente, os usuários precisam compartilhar uma série de confirmações em vez de um único conjunto de alterações. Em vez de confirmar um conjunto de alterações de uma cópia ativa para o repositório central, o Git permite compartilhar ramificações inteiras entre os repositórios.

O comando git remote é uma parte do sistema mais amplo que é responsável pela sincronização de alterações. Os registros com o comando git remote são usados junto com os comandos git fetch, git push e git pull. Todos esses comandos têm suas próprias responsabilidades de sincronização

O comando git remote permite criar, ver e excluir conexões com outros repositórios. As conexões remotas são mais parecidas com marcadores em vez de links diretos para outros repositórios. Em vez de dar acesso em tempo real a outro repositório, eles funcionam como nomes convenientes que podem ser usados para fazer referência a uma URL não tão conveniente.

Por exemplo, o diagrama a seguir mostra duas conexões remotas do seu repositório com o repositório central e o repositório de outro desenvolvedor. Em vez de fazer referência a eles pelas suas URLs completas, você pode passar a origem e os atalhos de John para outros comandos do Git.



O que e o GITHUB

Plataforma de hospedagem de códigos para controle de versão com o GIT, e colaborações

- Comunidade ativa
- Utilizado mundialmente
- Mascote “Octocat”

Histórico do GITHUB

- 2008 Desenvolvido por Cris Wanstrath, J. Hyett, Tom Preston-Werner e Scott Chacon.
- 2018 a plataforma foi vitima de uns doas maiores ataques de DDoS (ataque distribuído de navegação de serviços); foi comprado pela **Microsoft Corporation** por US$ 7,5 bilhoes.

Criando e Clonando Repositórios

Existem duas formas de obter repositórios na sua máquina

- Transformando um diretório (pasta) local que não está sobre controle de versão, num repositório GIT usando o comando **git unit.**
- Clonar um repositório GIT existente.

Adicionando e preparando arquivos pro commit

- **git Status** pra verificar documentos novos ou modificados na pasta
- git add “nome do arquivo” que queira adicionar ao diretório remoto
- git add . para adicionar todos os novos de uma só vez

Adicionado arquivos e pastas para o push

- **git commit** -m “Descrição do Commit”
- **git commit** “nome do arquvo.extensão” -m “Descrição do Commit”
- g**it log** pra verificar os commits existentes.