

RELATÓRIO DO VÍDEO

“SEJA LIVRE, USE SOFTWARE LIVRE – GIT”

Aluno: Antônio Marcos Lins da Silva Filho

Professor: Nilton Jose Rizzo

INTRODUÇÃO

O git é um sistema controle de versão, essencial para engenheiros de software, acomodando mais de 25 milhões de usuários, o uso do GIT pode melhorar o fluxo de trabalho e a colaboração. O serviço é baseado em nuvem que hospeda um sistema que permite que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do progresso.

Seu nome foi dado por uma gíria britânica para “cabeça dura”, pessoas que acham que tem sempre razão, segundo Linus Torvalds e seus serviços gratuitos são GitHub, Gitlab e bitbucket.

DESENVOLVIMENTO.

O Gitlab veio concorrendo com o github por sua gratuidade para um conjunto de repositórios públicos, permita que os repositórios também, diferente do Github, seus níveis de acesso são: público onde qualquer um pode acessar seu repositório, particular, o qual somente pessoas com conta no gitlab podem acessar e privado que somente o usuário tem acesso

Em comparação ao CVS e SUBVERSION, o GIT é mais vantajoso de se usar por sua facilidade, porém seu grande diferencial é que enquanto os outros marcam diferenças entre os arquivos, o git, mantém uma cópia total das versões anteriores, dos arquivos salvos ou alterados.

Existe também a opção de gerar chave SSH e PGP Key, que estabelecem uma conexão segura entre o computador e a plataforma gitlab, podendo entrar na plataforma de qualquer computador e ser reconhecido, evitando assim que pessoas indesejadas entrem sem autorização.

As ferramentas que o GIT possui, são:

- Repositório – Local onde são armazenados os arquivos e históricos de modificações, podendo ser local ou remoto.
- Snapshot – Congelamento do repositório

- Commit – Conjunto de modificações que vai armazenar em seu repositório, é quando ele vai criar um snapshot, ele salva as informações das alterações. É uma das ferramentas mais utilizadas
- Clonar - Comando para criar uma cópia do repositório remoto para o local.
- Pull – Comando de atualização, faz uma cópia do progresso do remoto para o local.
- Push – Comando de atualização, faz uma cópia do local para o remoto.
- Branches – Nomes dados a sequência de caminhos de commits, usualmente master sendo a sequência principal.
- Tags – Nomes dados a um commit específico
- Merge – Junção da alteração de uma branch na master
- Checkout – Quando precisamos reverter algo que foi feito.
- Fork – Duplicação do repositório de uma pessoa para podermos contribuir com seu projeto, sendo necessária a aprovação para fazer parte do projeto principal.
- Merge request – É a devolução da principal nossa alteração do projeto principal de outro.

As ferramentas gratuitas Git, que é linha de comando. Git Shell, simula shell para ser usado como git – cola, é um git com interface gráfica.

Após essa apresentação teórica começa a prática, onde primeiramente faz uma versão mais simples de “olá mundo”. Criou-se o diretório, mas não um repositório, para tal, é necessário dar o comando git init, para criar o repositório local. Adicionando o diretório no repositório com git add, tendo agora uma branch master e possibilitando o git commit, obrigatoriamente criando um comentário,

Após a remoção de um arquivo fez-se o checkout para obtê-lo novamente.

Criou-se uma branch chamada bugfix nessa branch adicionou-se, “hello world” nessa branch e voltou para a branch master.

Em seguida foi ensinado a criar um arquivo para ignorar algo indesejado, na formação das versões (gitignore).

CONCLUSÃO

Podemos concluir que o Git é de grande importância para programadores, pois nele além de conseguir visualizar todas as versões é possível que receba ajuda para resolução de bugs, por exemplo, e no desenvolvimento. Tem-se uma facilidade em seu uso e de organização.