FACULDADE DE TECNOLOGIA SENAC DE GOIÁS
CURSO GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

AUDITORIA E QUALIDADE SOFTWARE

Henrique Alberto Morato
João Paulo Nascimento Oliveira
Paulo Roberto Vieira
Tarcisio Lopes Albernaz Gomes

Controle de versão

Goiânia
2020

Para realizar esta demanda é necessário ter o Git instalado em uma máquina e também ter conta/usuário no GitHub.

- **Git**

É sempre recomendado que utilizem a documentação oficial de uma ferramenta se tratando da área de tecnologia da informação, neste caso sugerimos para conhecer mais sobre o Git, a página (https://git-scm.com). De forma bem resumida GIT é um sistema de controle de versões que registra alterações em um arquivo ou conjunto de arquivos ao longo do tempo, caso seja necessário você pode recuperar uma versão especifica “na linha do tempo”. Aqui abordaremos especificamente o versionamento de software, mas o Git pode ser usado basicamente com qualquer tipo de arquivo no computador (CHACON; STRAUB, 2020). Devemos ressaltar que ao utilizar Git basicamente o versionamento ocorre na máquina que está em uso, para versionamento web é outra ferramenta, no caso o repositório GitHub, porém é comum o aprendizado e uso das duas ferramentas de forma conjunta.

- **GitHub**

O GitHub é um repositório de hospedagem Git que fornece aos desenvolvedores ferramentas para enviar código melhor através de recursos de linha de comando, problemas (discussões encadeadas), solicitações de recebimento, revisão de código ou o uso de uma coleção de aplicativos gratuitos, Github (2020).

Trabalhar em repositórios mantém os projetos de desenvolvimento organizados e protegidos. Os desenvolvedores são incentivados a corrigir bugs ou criar novos recursos, sem medo de inviabilizar os esforços de desenvolvimento da linha principal. O Git facilita isso por meio do uso de ramificações de tópicos: indicadores leves para confirmações no histórico que podem ser facilmente criadas e reprovadas quando não forem mais necessárias Github (2020).

Por meio de plataformas como o GitHub, o Git também oferece mais oportunidades de transparência e colaboração em projetos. Repositórios públicos ajudam as equipes a trabalharem juntas para criar o melhor produto final possível Github (2020).

- **Usando GitHub**

Primeiramente deve-se criar um usuário para ter acesso ao repositório GitHub, na página (https://github.com) é possível criar uma conta informando Username, Email e Password.

Com acesso ao sistema, o usuário já consegue criar um repositório (Create repositor ou Repositories - New) ou mesmo importar um já existente. Com um repositório é possível criar arquivos localmente, porém o mais comum é a "transferência" de arquivos de uma máquina utilizando o Git, para tal abordaremos o Git e seus comandos a seguir.

- **Usando Git**

Para usar o Git em sua máquina é necessário que seja feita a instalação, dependendo do sistema operacional utilizado o modo de instalar irá variar, mas é possível encontrar várias referências na web de como proceder com a instalação, a seguir, veja onde conseguir instalador ou comando para instalar.

Windows - https://git-for-windows.github.io/

Linux (verificar o gerenciador de pacotes do S.O) - (Ubuntu) apt install git

Mac - http://sourceforge.net/projects/git-osx-installer

- **Configurações**

Após efetuar instalação, algumas configurações são essenciais, demonstraremos abaixo as configurações e os comandos básicos para se iniciar no Git e GitHub.

- **Configure seu Usuário**

O controle de versão controla quem faz as alterações num projeto. Então você deve configurar seu usuário:

git config --global user.name "AlunoSenac"

git config --global user.email "seu.email@gmail.com"

- **Crie um Repositório**

Primeiro crie uma pasta e dentro dela execute o seguinte comando:

git init

O repositório é onde estarão os arquivos versionados, ou seja, dentro da pasta criada haverá um repositório git.

- **Adicionado e comitando**

Estando dentro da pasta criada e o repositório tendo sido iniciado. Você deve criar um arquivo qualquer, utilize o editor de texto de sua preferência.

Após criar um arquivo e edita-lo vamos adiciona-lo ao repositório.

git add nomedoarquivocriado+extensão (git add MediaFinal.c)

git commit -m "comentário para identificar commit" (git commit -m “Programa Calcular média”)

- **Conectando o repositório local com o da web**

O seu Git é o seu repositório local. A partir dele, você enviará ou receberá informações do repositório na web, do GitHub.

Para “conectar” os repositórios vá no GitHub e copia a url de seu repositório e posteriormente efetue o seguinte comando na linha de comando do Git.

git remote add origin AQUI VAI A URL

Onde origin é um apelido para seu repositório, poderia ser qualquer outro nome.

Ex. git remote add ProjetoIntegrador

https://github.com/seurepositorio/ProjetoIntegrador.git

- **Sincronizando os repositórios**

Tendo criado repositório local, outro no GitHub e adicionado alguns arquivos e “comitou” algumas modificações (localmente). O seu repositório local (Git) já está conectado com o da web (GitHub). Agora falta enviarmos as informações do repositório local para o repositório na web.

git push ProjetoIntegrador master

- **Projeto Integrador**

Com o git instalado, conta criada no github, basicamente a partir deste ponto é usar alguns comandos, trabalhamos com uma conta e somente no branch master, portanto não há outros branchs criado neste projeto. De forma bem resumida e simples, utilizamos:

adicionar e confirmar

Você pode propor mudanças (adicioná-las ao Index) usando

git add <arquivo> ou git add *

Para realmente confirmar estas mudanças, ou seja, fazer um commit, use

git commit -m "comentários das alterações"

Agora o arquivo é enviado para o HEAD, mas ainda não para o repositório remoto, ou seja, ela ainda está local.

- **Enviando alterações**

Com as alterações no HEAD, aplicamos o push e assim enviamos estas alterações ao repositório remoto.

git push origin master

- **Atualizar e mesclar**

Para atualizar o repositório local com a mais nova versão, executamos

git pull

na pasta de trabalho para obter e fazer merge (mesclar) alterações remotas.

Referências

CHACON, Scott; STRAUB, Ben. **Pro Git Book,** 2.ed. 2020. Disponível em: https://git-scm.com/book/en/v2. Acesso em: 14 maio 2020.

GITHUB. **GitHub Guides**: Git Handbook. 2020. Disponível em: https://guides.github.com/introduction/git-handbook/. Acesso em: 14 maio 2020.

Anexo I – Principais Comandos (Retirado Slides Professor Luciéliton Mundim)

Para transformar um diretório local em um repositório Git, basta abrir o diretório via terminal e executar o comando:
git init
ou
git init repositorio_teste

Supondo que adicionamos o arquivo filmes.txt no diretório "repositorio_teste"
git status

Para que um arquivo seja rastreado, devemos executar o seguinte comando:
git add filmes.txt ou
git add * (para incluir todos os arquivos)

Para gravar as mudanças no repositório, execute o comando:
git commit -m "Arquivo inicial de filmes para download"

Para visualizar o histórico de commits
git log

Para mostrar as alterações de um commit
git show [commit]
Commit deve ser especificado pela chave

Para listar as branches do nosso repositório, devemos executar o comando:
git branch

Como listar todas as branches?
git branch -v (- r para branch remota, ver depois)

Como criar uma branch?
git branch nome_da_branch

Modo mais rápido de criar e deixá-la como corrente:
git checkout -b nome_da_branch

Apagar uma branch em seu repositório
git branch -d nome_da_branch

Como mudo de branch?
git checkout nome_da_branch

Apagar uma branch remota
git push origin --delete nome_da_branch
Visualizar logs dos commits
git log
git log --graph --oneline

Verificar diferenças entre commits
git diff 3de30bc a810c6a

Remote
• Repositório remoto, hospedado em um servidor
• São referenciados por uma URL
• Podem receber vários commits
• Sincronizar o trabalho colaborativo
• Exige chave SSH
git remote add origin <URL>

Push
• Enviar alterações (commits) de uma branch para o repositório remoto
• A primeira vez:
git push -u origin master
• O envio é rejeitado se o repositório local não estiver sincronizado
git push <remote> <branch>
git push

Pull
• Baixa as alterações do repositório remoto e realiza o Merge automático com o repositório local
• Mantém o repositório sincronizado com os últimos commits de uma branch
• Permite baixar novas branchs que não foram criadas localmente
git pull
git pull <remote> <nova>

Git Clone
• Baixa o repositório remoto
• Outra forma de criar um repositório local
• Já vem com o remote configurado
git clone <URL>