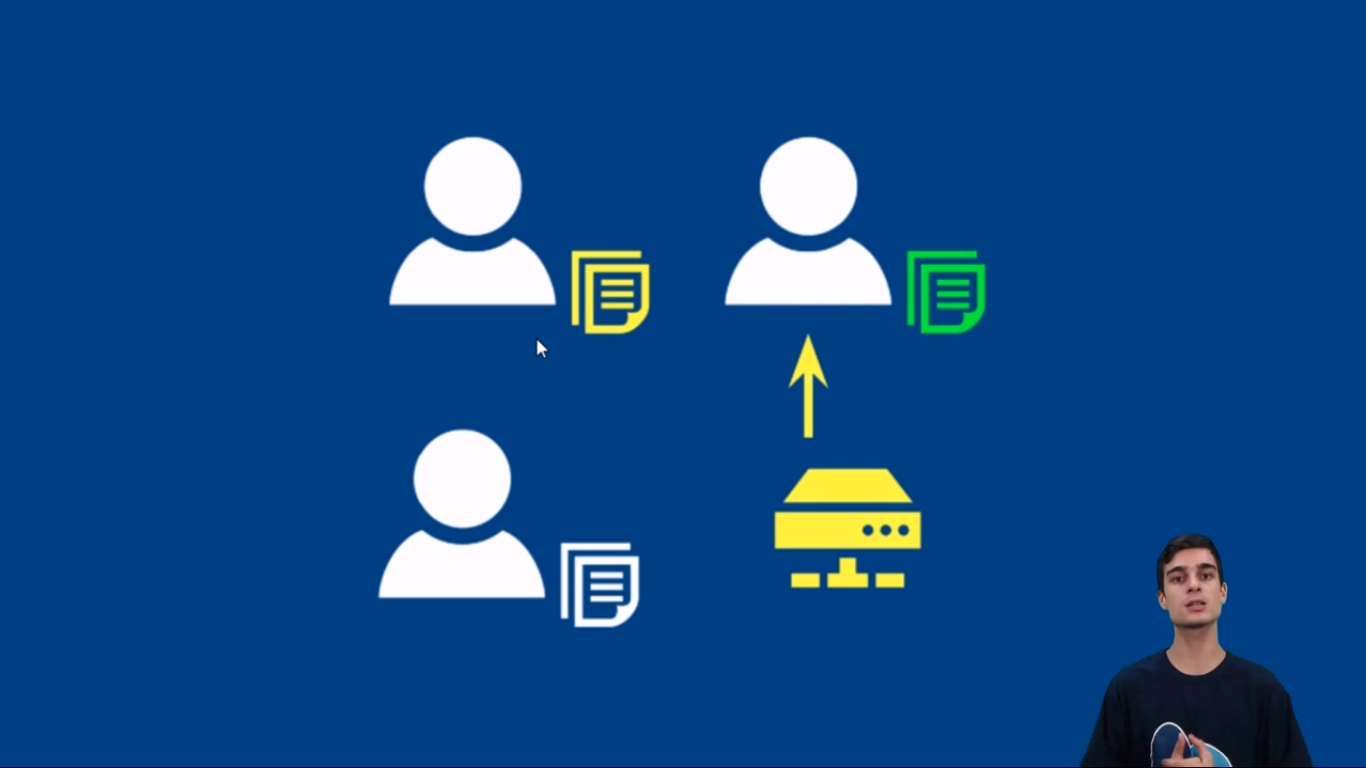
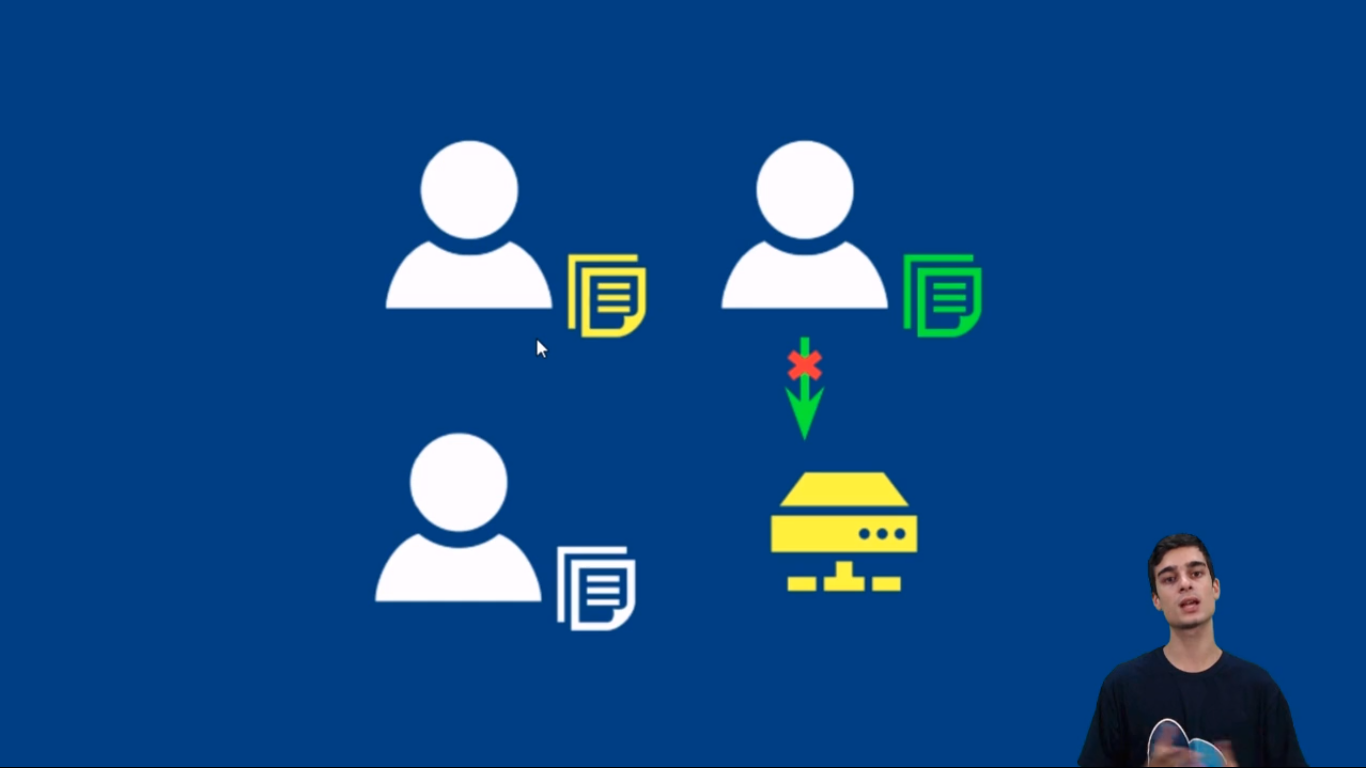
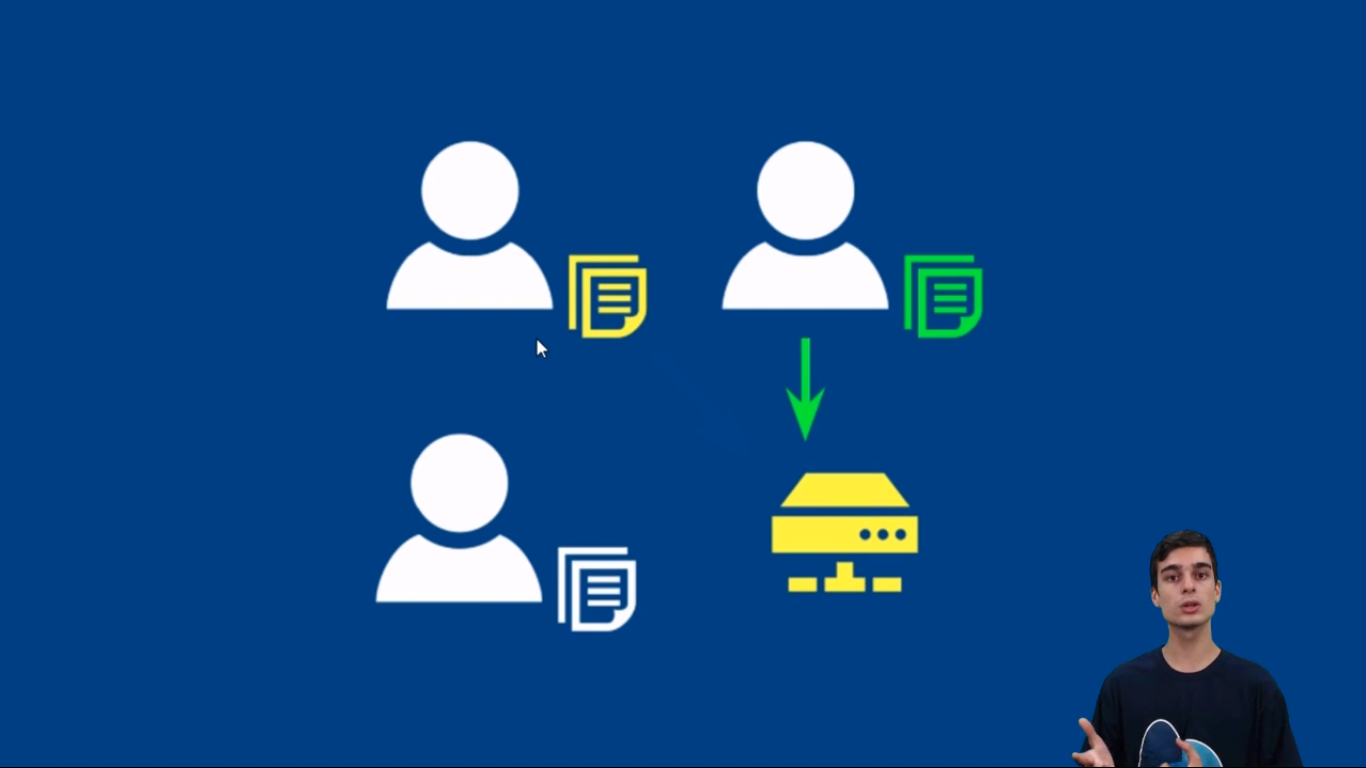
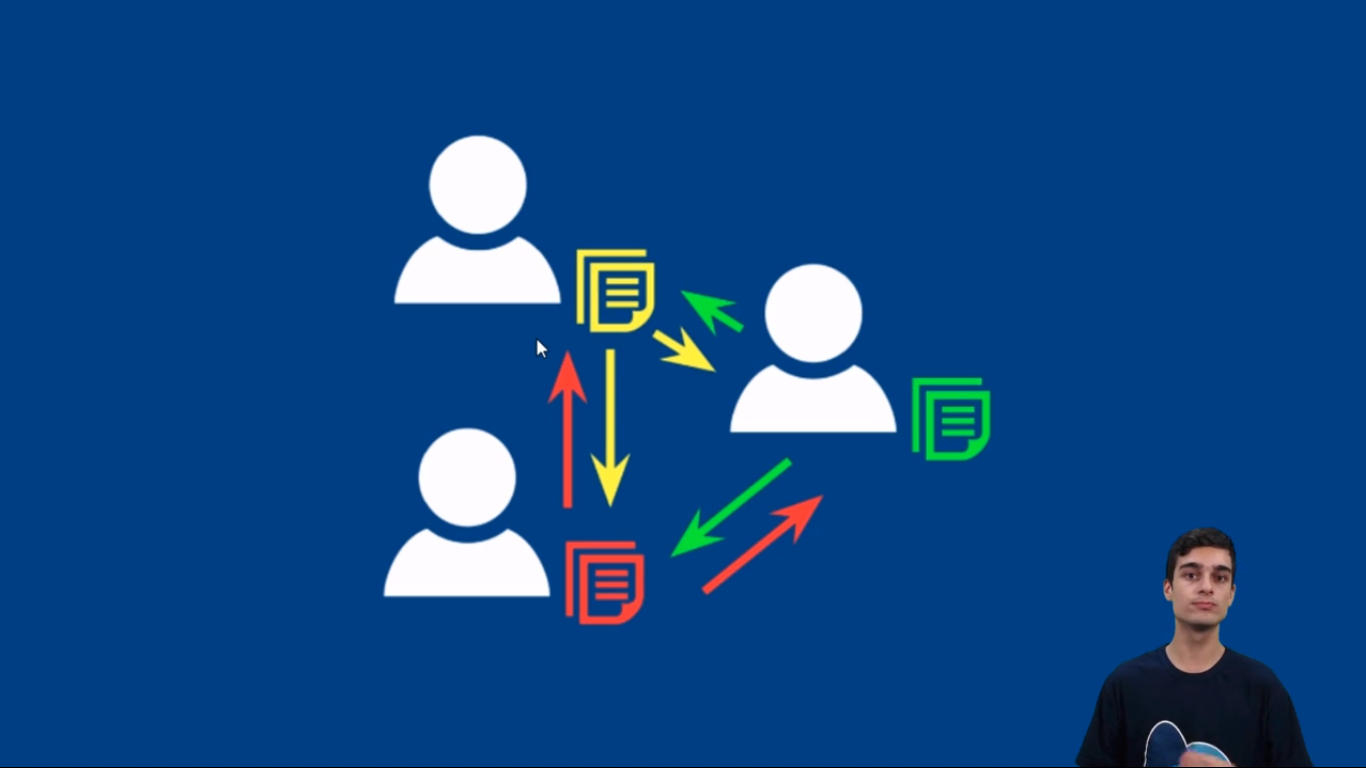
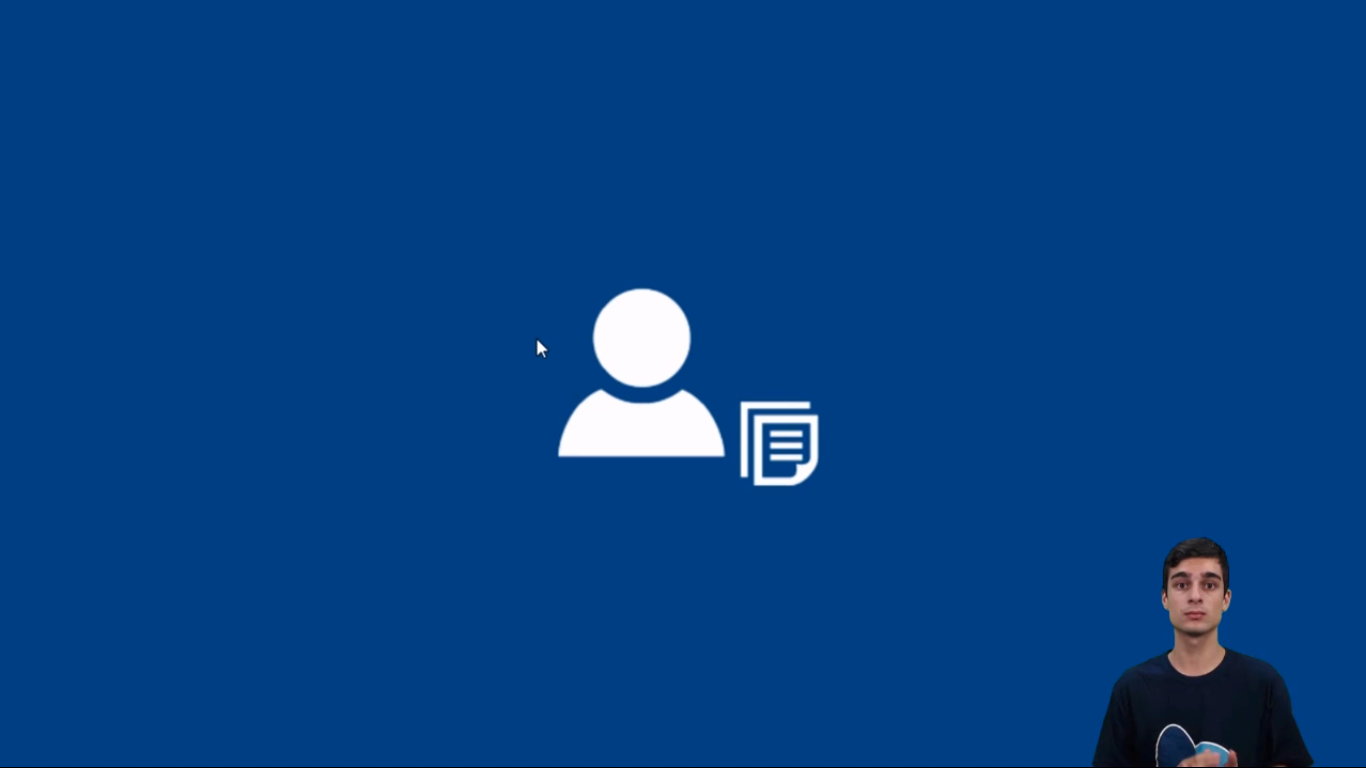
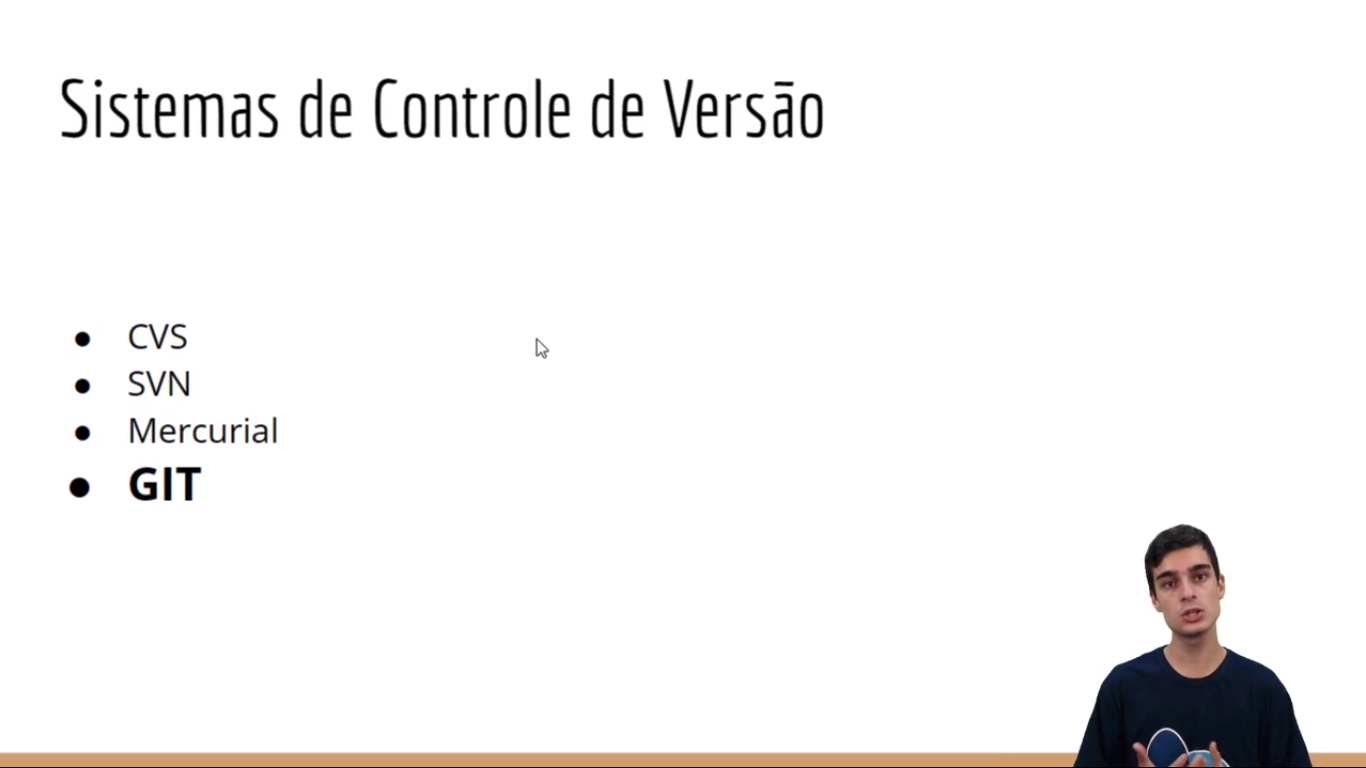
Semana 06

Git e Github - Controle e compartilhe seu código





Baixe e instale o git em sua máquina

git-scm.com

ao utilizar o git em sua máquina você poderá usar tanto o terminal do Windows (cmd) quando o terminal do git(git bash), nos próximos exemplos iremos utilizar o git bash

ls - lista os arquivos

git --version - verifica a versão instalada em sua máquina

iremos usar o repositório como exemplo:

/Documents/git-e-github/vinicius

Dentro desse repositório teremos o arquivo index.html e digitaremos

git init – inicia um novo repositório



Ao final do caminho será mostrado o ramo que se encontra esse repositório, nesse caso master

Para informar ao git quem sera o usuário:

git config --local user.name "Seu nome aqui"

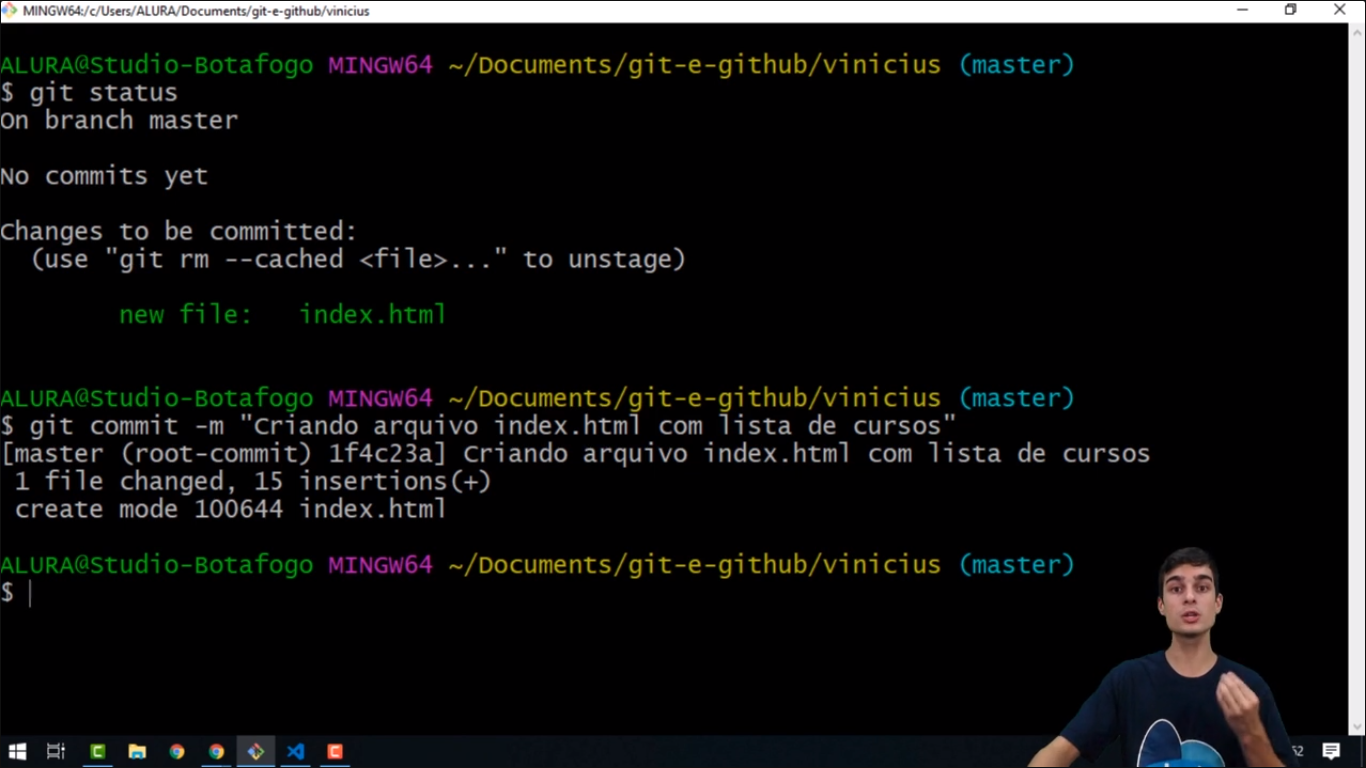
git config --local user.email "seu@email.aqui"

git add index.js – para monitora qualquer mudança nesse arquivo

git add . – para monitora todos os arquivo nesse caminho

git rm --cacher <arquivo> - para cancelar o monitoramento de um arquivo

git commit -m “primeiro commit” – enviar para área de commitado



git log – lista os commits feitos um históricos

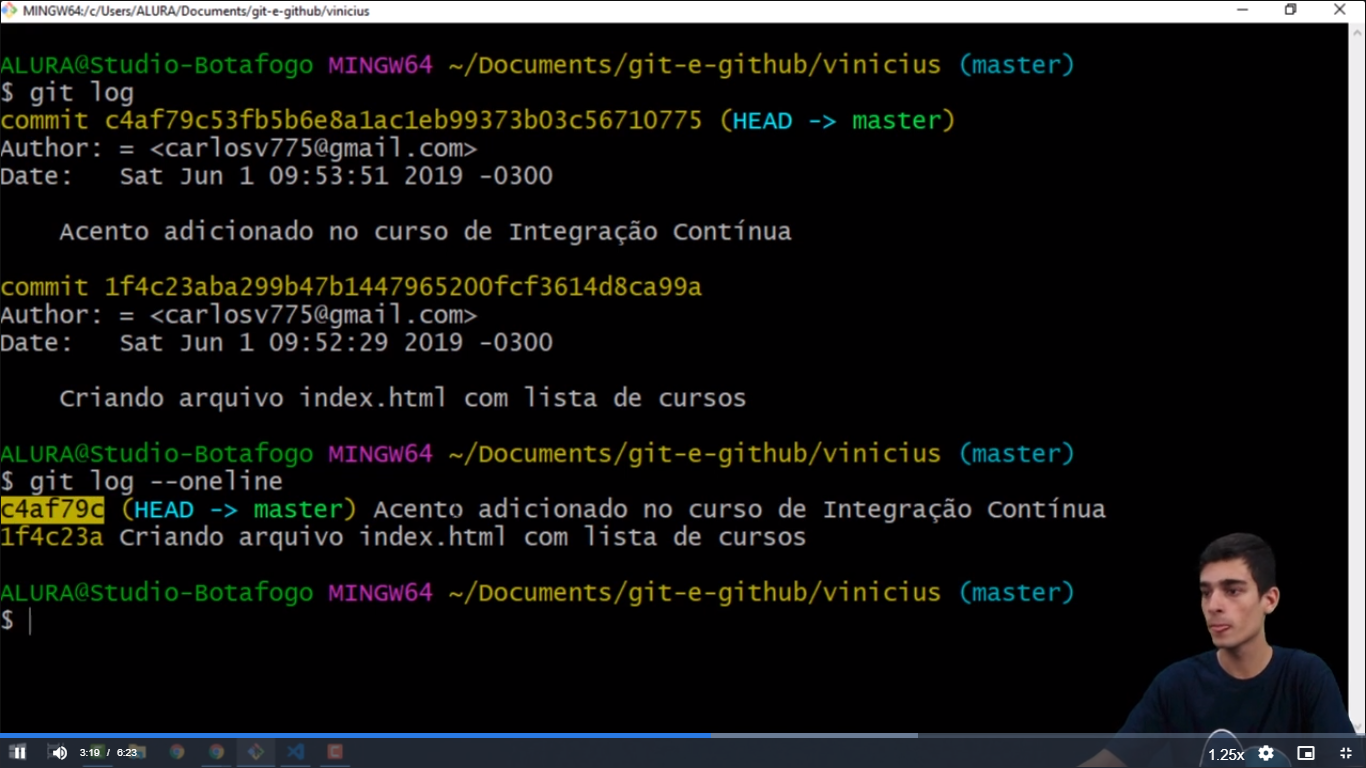
git config --local – para configurar somente para esse projeto

git config --global – uma configuração para todos os projetos

git config --local user.name “rodrigo de souza” – configura a configuração desse projeto como nome do usuário sendo rodrigo de Souza

git config user.name – para verificar qual o nome está no projeto

git log --oneline – mostra o log resumido em uma linha mostrando apenas o hash o branch que está e sua mensagem



git log -p – mostra as informações de cada log mais detalhada

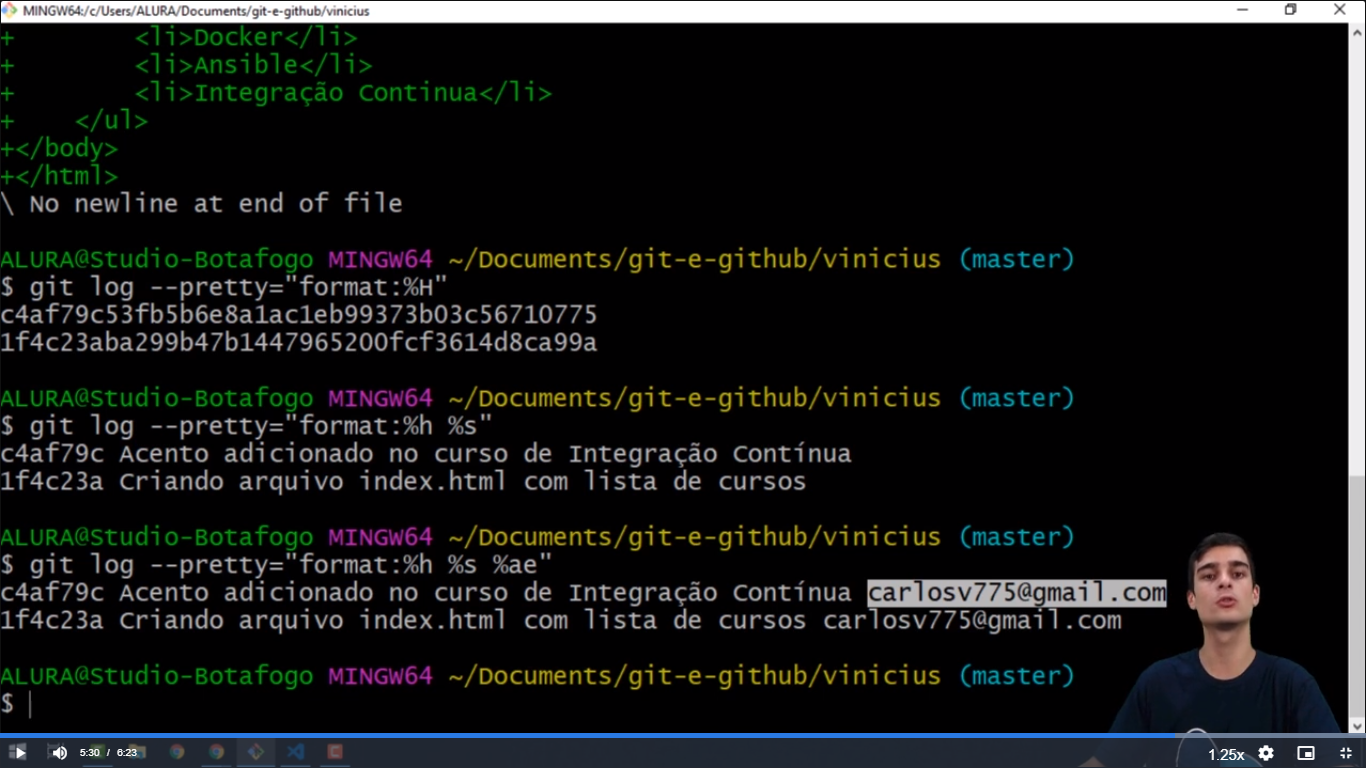
link para algumas opções que podem ser usada para mostrar o log

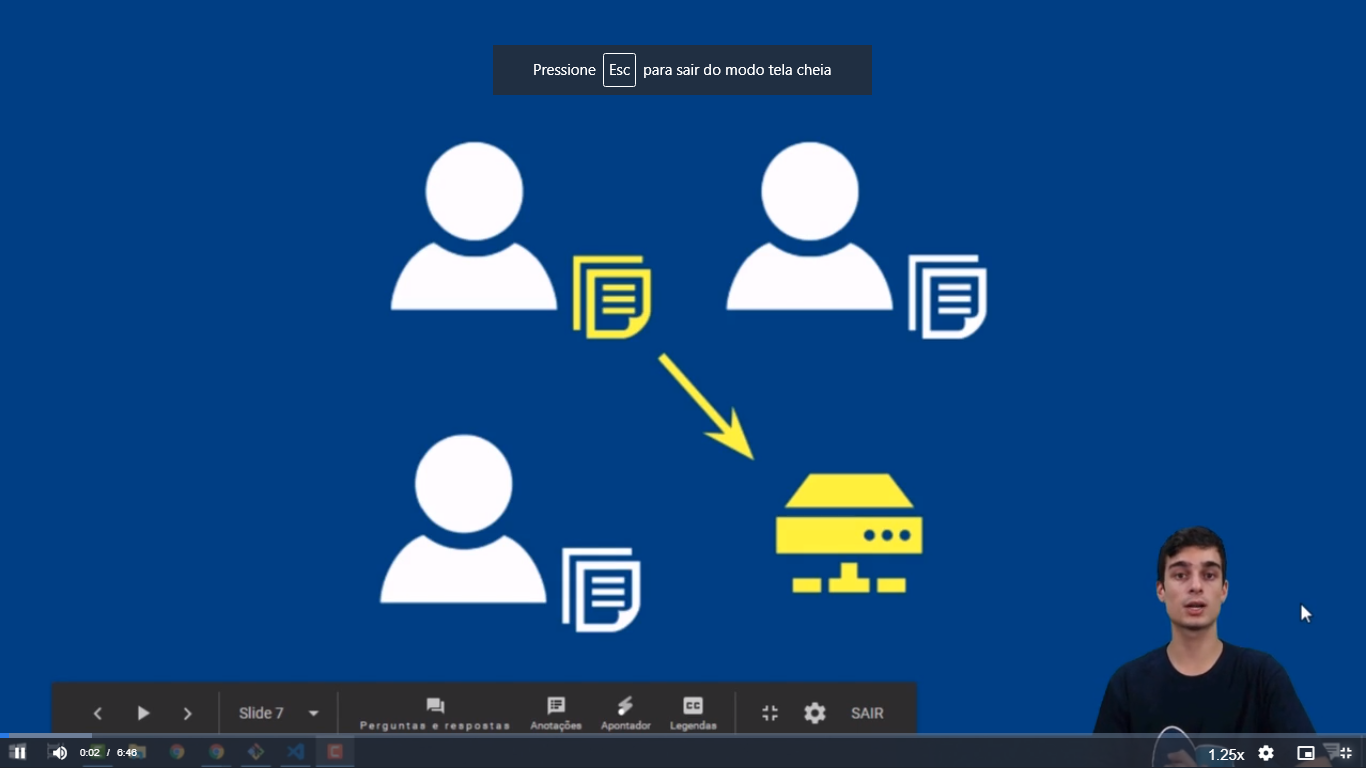
<https://devhints.io/git-log>.

git log --pretty=”format:%H” – para formata o log mostra só o hash



git log --pretty=”format: %h %s %ae” – mostrar no log a hash mensagem e o autor no log



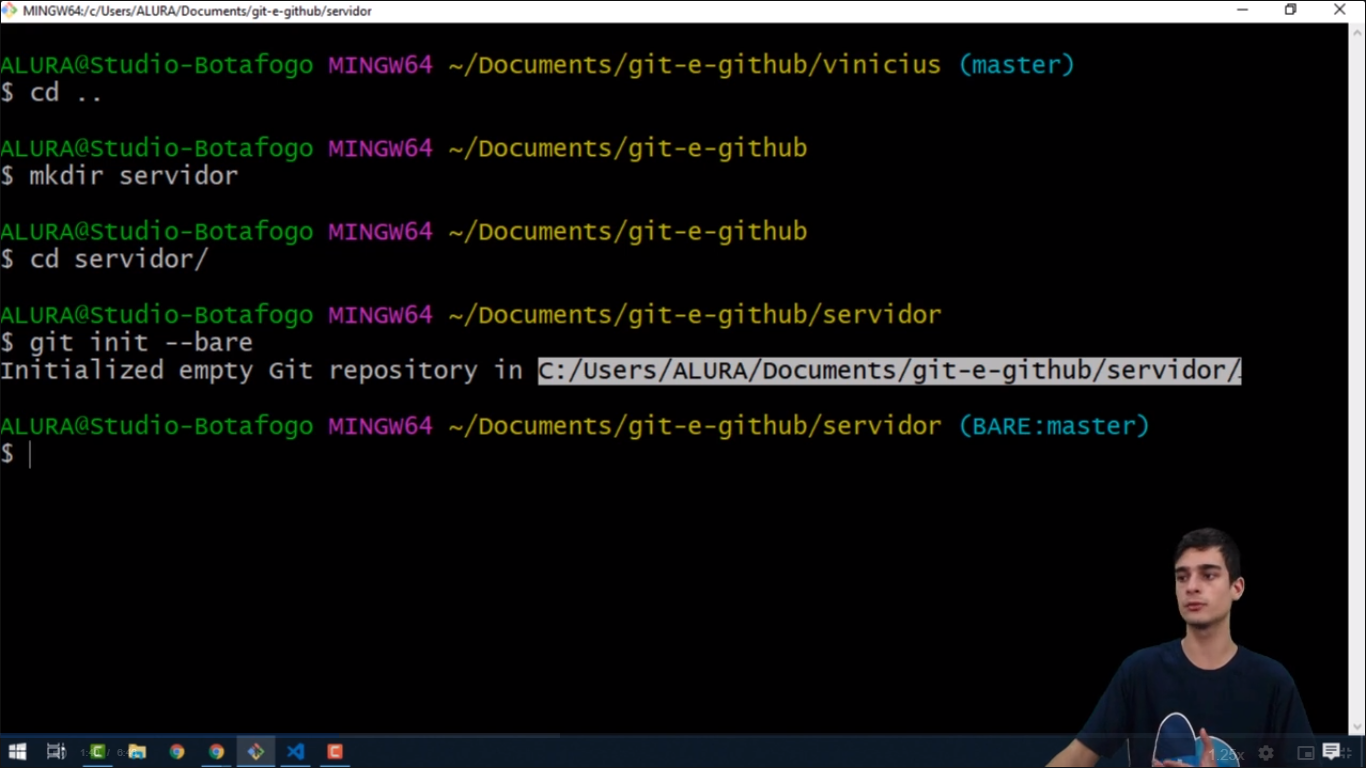


mk dir servidor – criado no caminho /Documents/git-e-github

irá criar uma pasta servidor nesse caminho

no /Documents/git-e-github/servidor foi feito um init

git init --bare – esse tipo de init significa que esse repositório só vai servir para armazenar informações



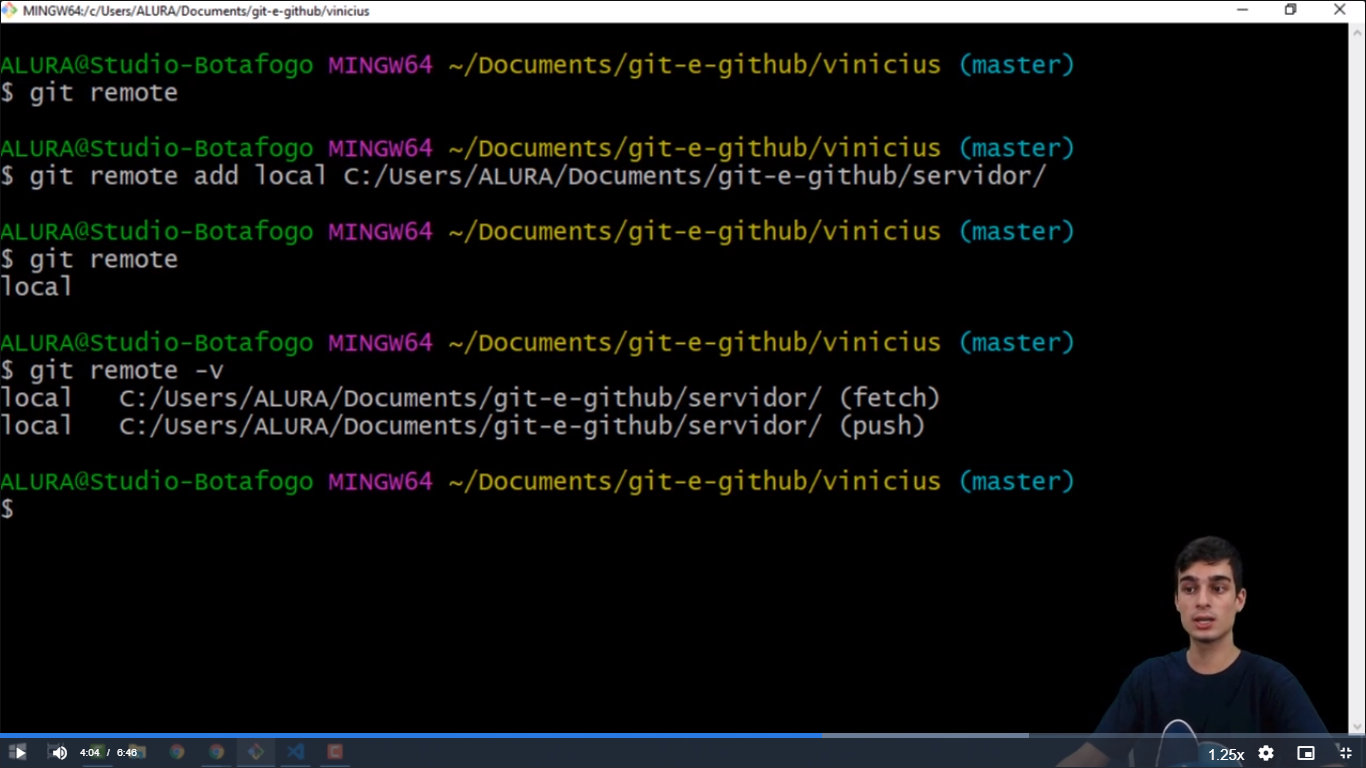
Voltando para o caminho do vinicius

git remote – lista todos os repositório remotos que o usuário local conhece

git remote add <nome\_repositorio\_remoto> <caminho\_para\_esse\_repositorio> - cria um novo repositório remoto

git remote add local c:/users/alura/documents/git-e-github/servidor/ - foi criado um rep. Local chamado local para o caminho indicado

git remote -v – mostra os endereços dos repositório remoto



Fetch – buscar dados nesse caminho

Push – enviar dados para esse caminho

mkdir ana – criado uma pasta para ana

documents/git-e-github/ana

git clone <caminho\_repositorio> - para copiar e clonar um repositório

git clone c:/users/alura/documents/git-e-github/servidor/ projeto - ira clonar os dados que estão nesse repositório com o nome projeto caso esse último parâmetro não existisse, esse projeto iria se chamar servidor