Introdução ao Git e ao Github

25/09

# Entendendo o que é Git e sua importância

GIT

* Sistema de versionamento de código distribuído, 2005, Linus Torvalds.

Aprendizados do curso

1. Controle de versão
2. Armazenamento em nuvem
3. Trabalho em equipe
4. Melhorar seu código
5. Reconhecimento

##### 

GIt Hub

* É um servidor remoto para o armazenamento de código.

Git e GitHub ajudam a trabalhar em equipe.

* Porque mostra de forma gráfica seu trabalho para outros desenvolvedores.
* Porque possui ferramentas que favorecem o desenvolvimento em equipe.
* Porque versiona seu código de forma segura e disponibiliza o mesmo na internet.

# Navegação via Command Line interface e instalação

GUI

* Interface de gráfica para usuário

CLI

* Interface de linha de comando

| Windows | Unix |
| --- | --- |
| cd (change directory) | cd |
| dir (directory) | ls |
| mkdir (make directory) | mkdir |
| rmdir (remove directory) | rm, -rf (r: recursive, f: force) |
| cd .. (Retorna ao diretório anterior) | cd .. (Retorna ao diretório anterior) |
| cls (clear screen) | clear, ctrl l |
| tab (autocompletar) | tab (autocompletar) |
| echo (retorna o texto digitado) | echo (retorna o texto digitado) |
| > (redirecionador de fluxo - atribuição) | > (redirecionador de fluxo - atribuição) |
| del (Exclui arquivos) |  |

Silence on sucess - O comando feito de maneira correta não traz retorno para o terminal.

# Entendendo como Git funciona por debaixo dos panos

SHA1

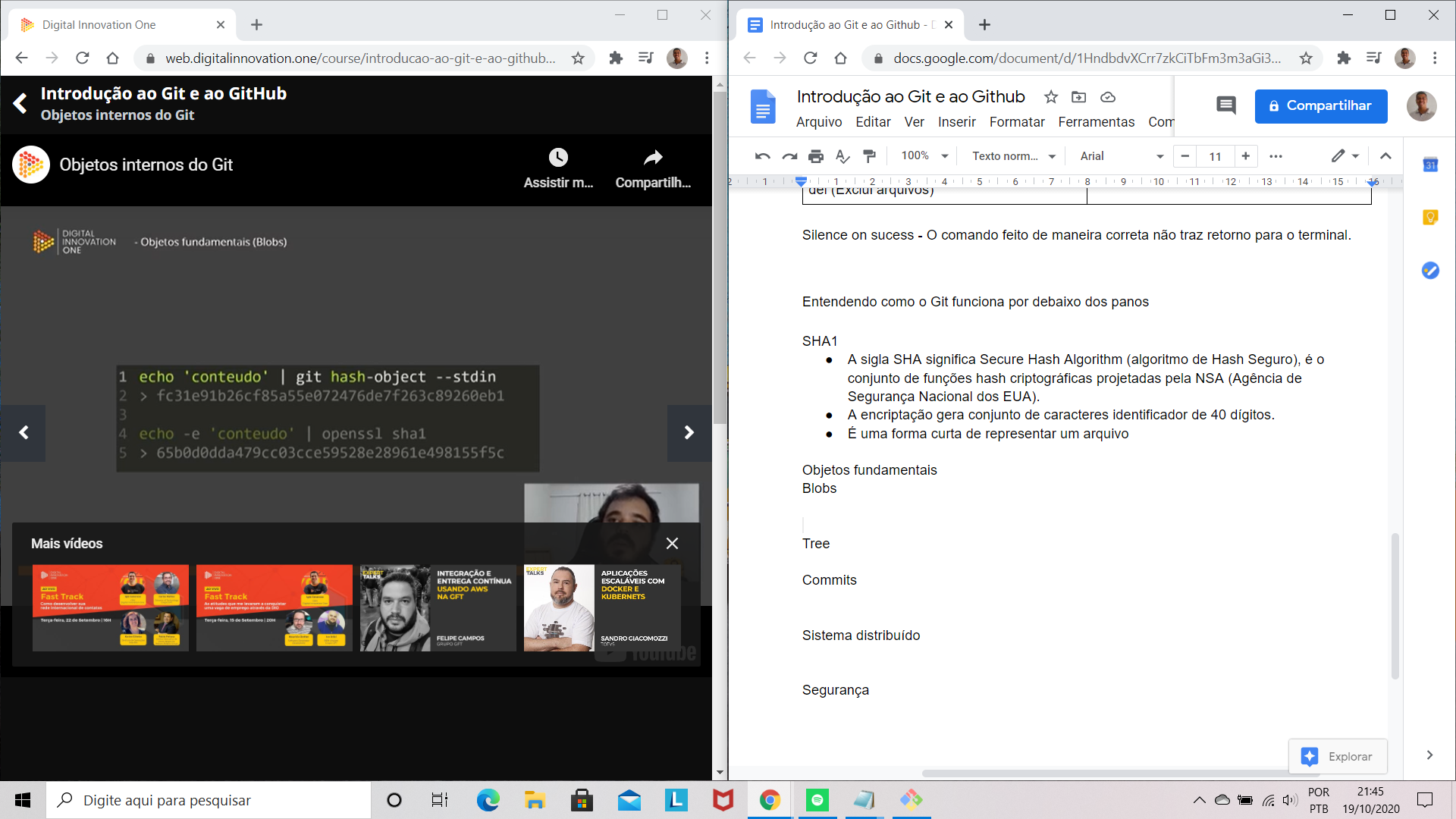
* A sigla SHA significa Secure Hash Algorithm (algoritmo de Hash Seguro), é o conjunto de funções hash criptográficas projetadas pela NSA (Agência de Segurança Nacional dos EUA).
* A encriptação gera conjunto de caracteres identificador de 40 dígitos.
* É uma forma curta de representar um arquivo

Objetos fundamentais

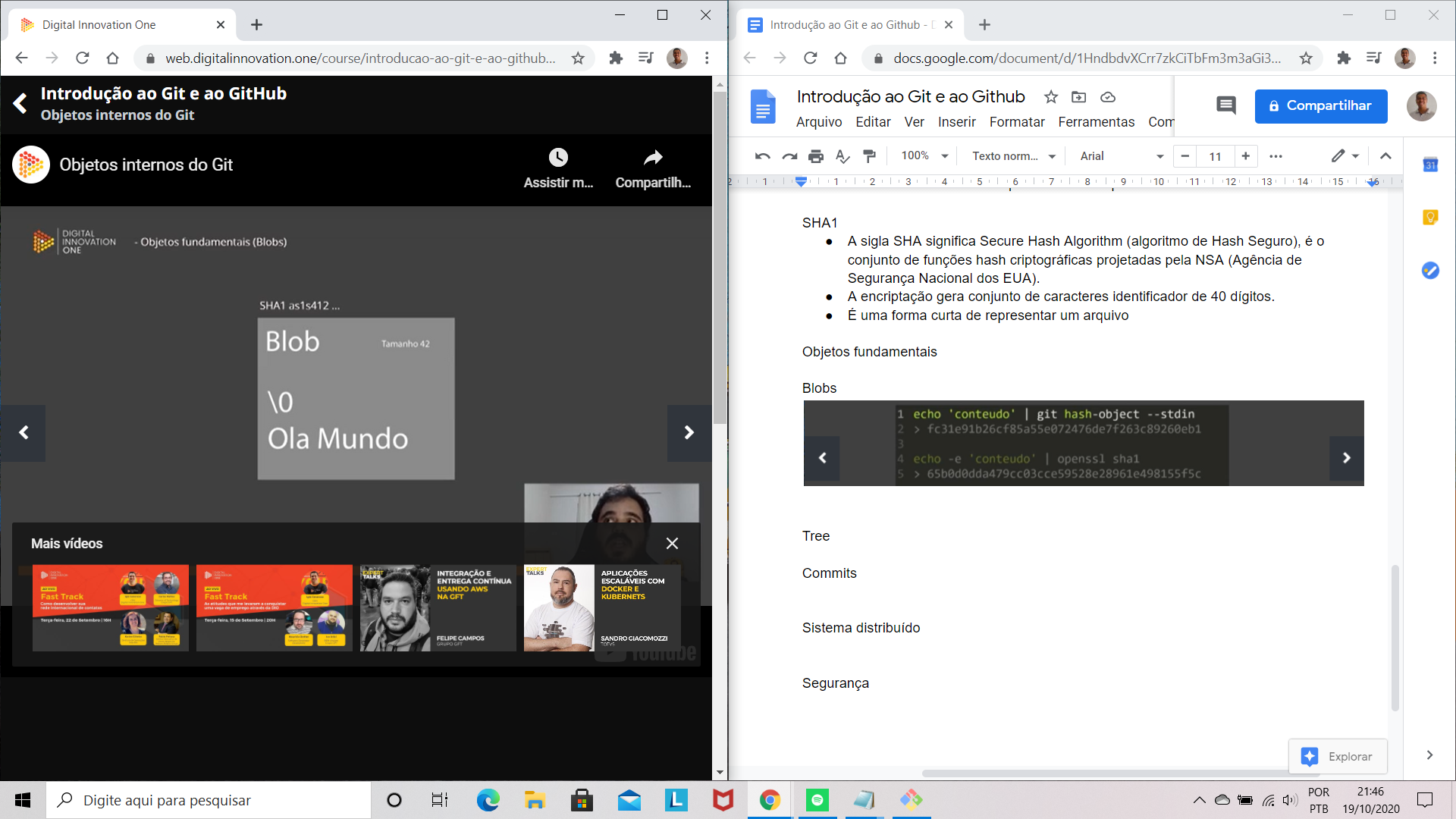
* BLOBS
* TREES
* COMMITS

Blobs

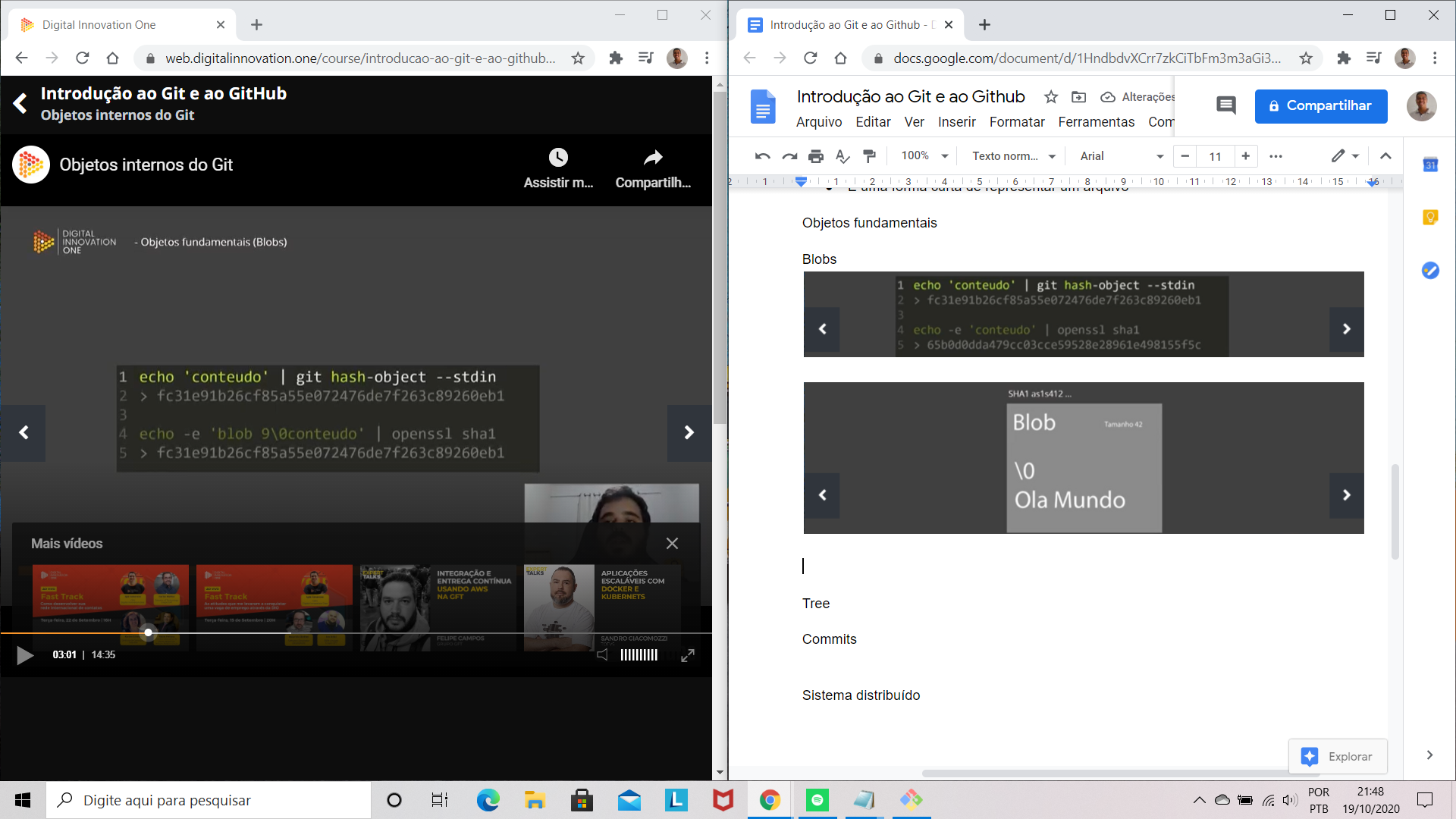
* Bloco básico de composição



* O GIT gera hash para cada operação
  + No exemplo, é solicitado o hash da string ‘conteudo’ de duas maneiras diferentes, uma pela flag hash-object e outra pelo openssl. O que resultou em dois hashis diferentes



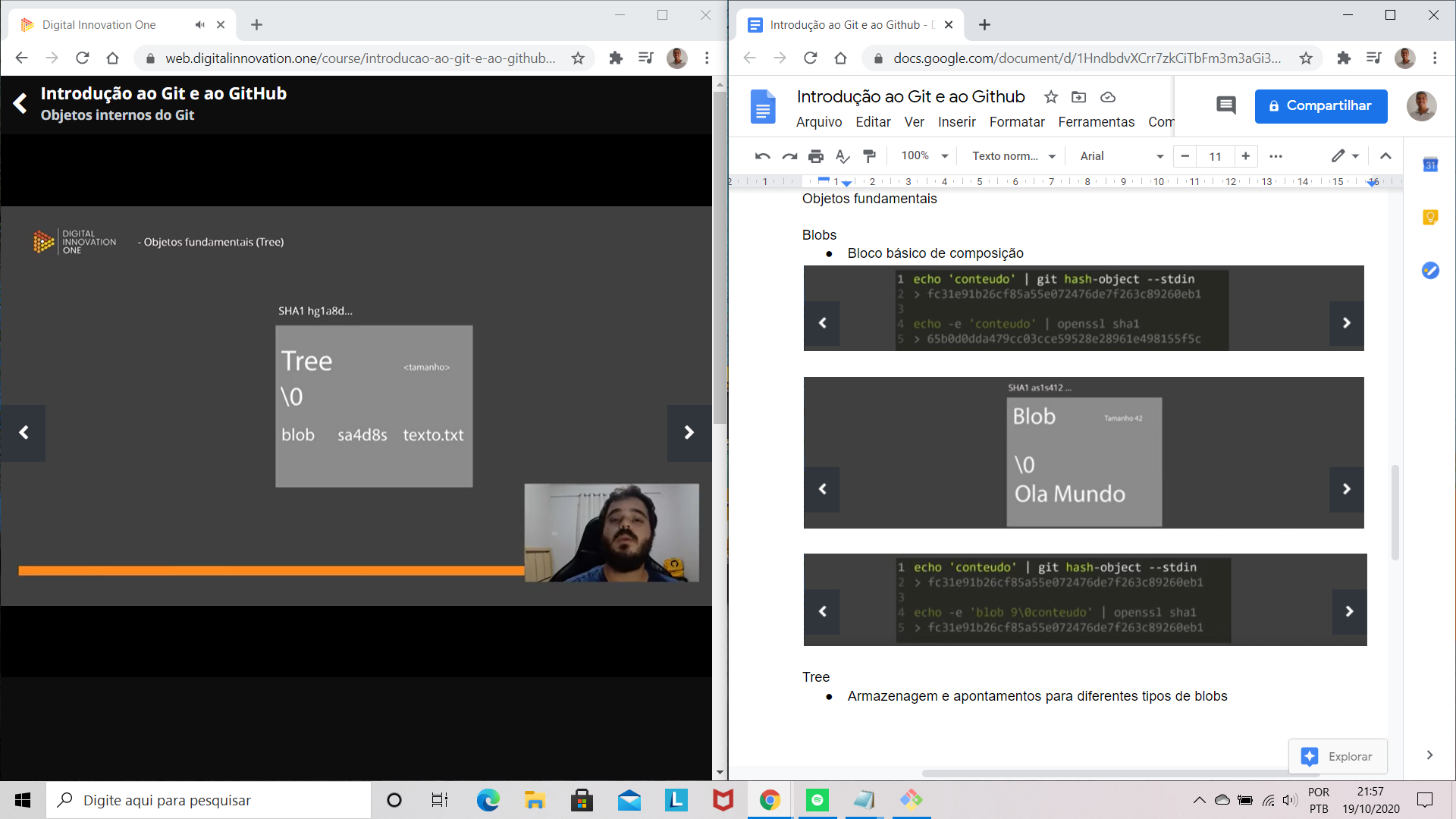
* Bloco básico de composição do GIT
* O BLOB é o objeto fundamental do GIT responsável por armazenar arquivos
* O BLOB possui os seguintes metadados
  + Tipo - Blob
  + Tamanho - 42
  + \0
  + Conteúdo - Olá Mundo



* Utilizando o openssl e passando os os metadados (tipo, tamanho, \0, conteúdo) obtemos o mesmo valor do hash

Tree

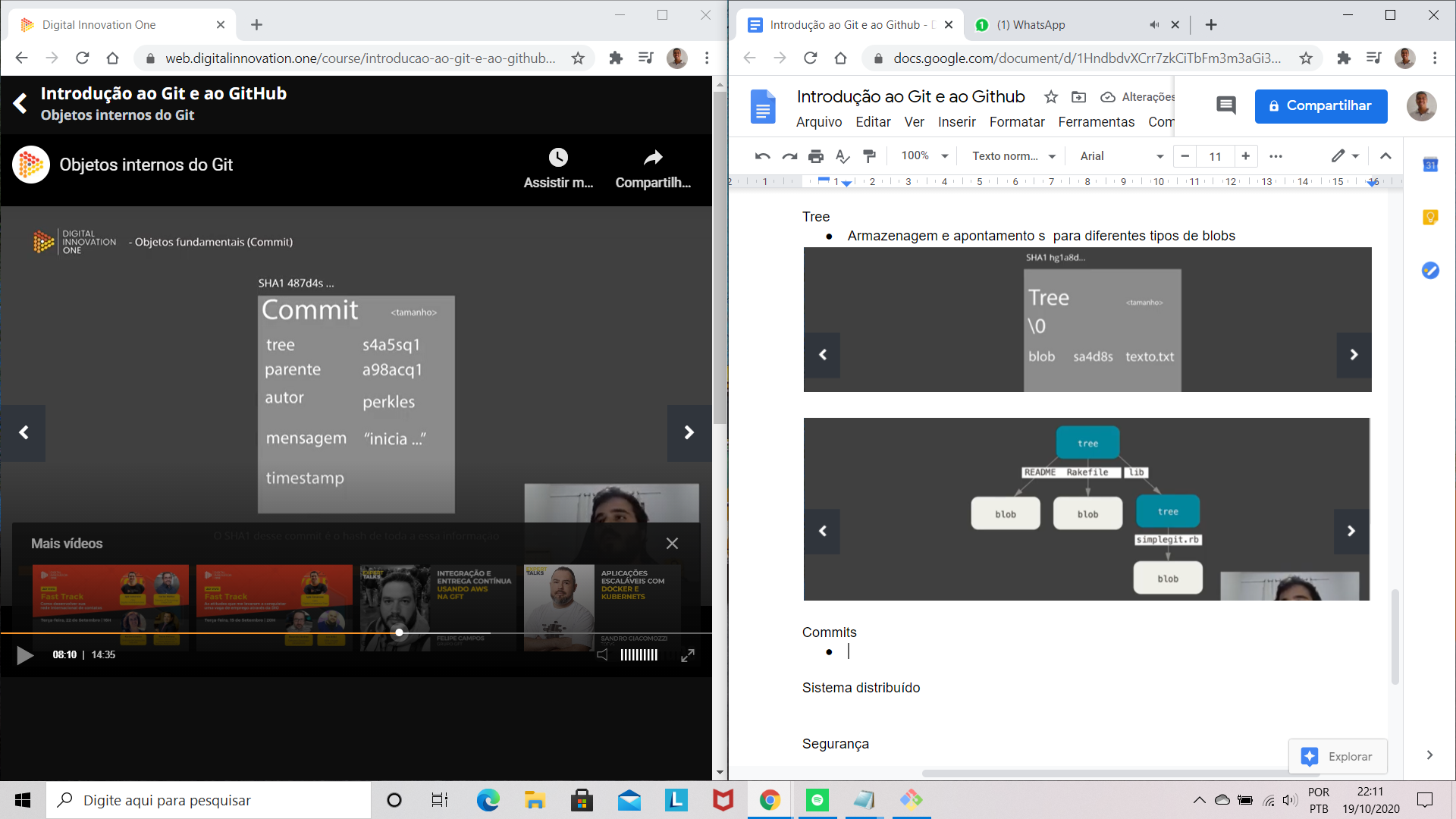
* Armazenagem e apontamentos para diferentes tipos de blobs
* Podem apontar para Blobs ou outras Trees



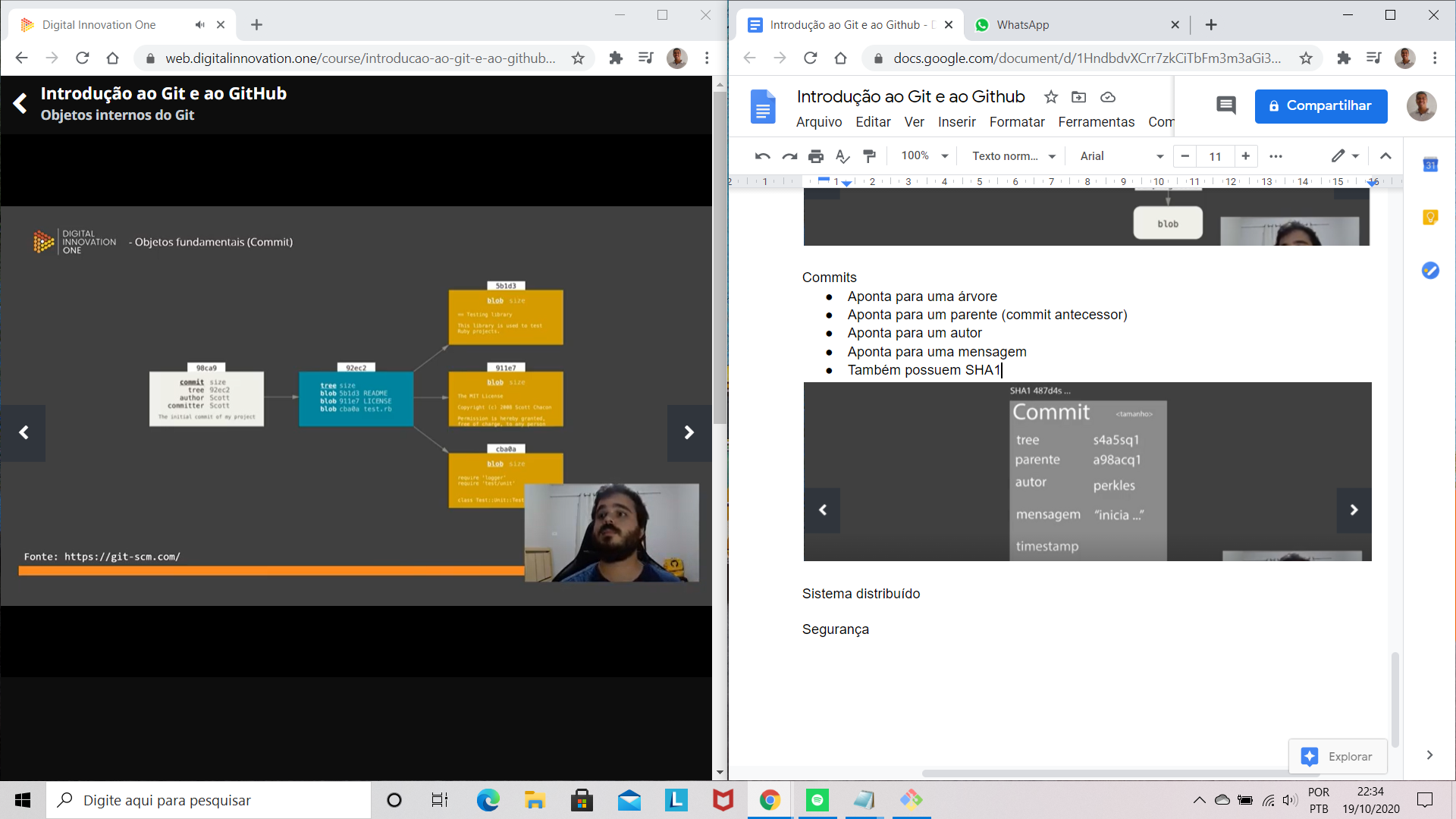
* Possui os seguintes metadados
  + Tipo - Tree
  + \0
  + Objeto de apontamento - Blob
  + Sha1 do objeto de apontamento - sa4d8s
  + Nome do arquivo - texto.txt



Commits



* Aponta para uma árvore
* Aponta para um parente (commit antecessor)
* Aponta para um autor
* Aponta para uma mensagem
* Timestamp - Carimbo de tempo do momento da criação
* Também possuem SHA1
  + O SHA1 desse commit é o hash de toda a essa informação

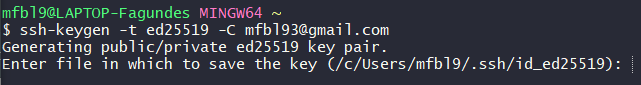


Sistema distribuído Seguro

* Versionamento confiável

Chaves SSH e Tokens

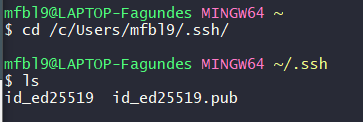
* Chave SSH - reconhecimento entre a máquina e o Github
* Para gerar a chave SSH devemos acessar o Github e seguir os seguintes passos
  + Acessar a guia Settings
  + Acessar a opção SSH e GPG Keys
    - Abrir o Git Bash
    - Aplicar os seguintes comandos:



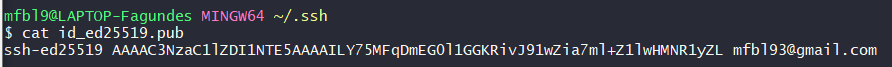
* + - Onde:
    - ssh-keygen - é o comando para geração da chave
    - ed25519 - é o tipo de criptografia da chave
    - [mfbl93@gmail.com](mailto:mfbl93@gmail.com) - e mail pessoal, de preferência o mesmo e mail usado no github
    - Como resposta o sistema mostra a localização da chave
    - Diretórios que são iniciados com “.” indicam que a pasta é oculta



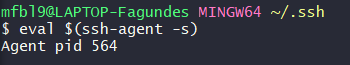
* + - Após pressionarmos enter
    - O sistema solicita uma senha
    - Após fornecermos a senha, os sistema mostra a localização da chave pública e privada
    - Ambas a chaves possuem o mesmo nome, porém a chave pública tem o .pub
    - O sistema também apresenta a fingerprint da chave

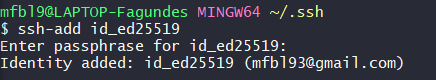


* + - Navegando até o local das chaves



* + - Usando o comando cat, conseguimos visualizar a chave publica
    - Devemos copiar ela e inserir na opção indicada no github: SSH Keys / Add new
    - No campo Título, inserimos o nome da máquina em que foi gerada a chave
    - Inserimos a assinatura da chave e realizamos a adição da chave
* Em seguida devemos fazer o processo de inicializar o SSH Agent



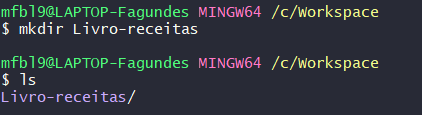


* Adicionando a chave privada para o agente

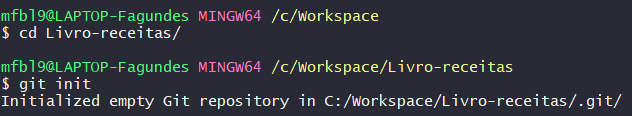
Tokens de acesso pessoal

# Primeiros comandos com GIT

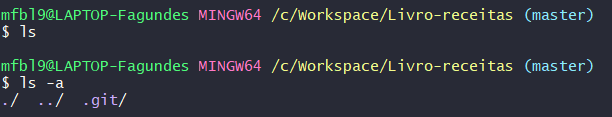
* Iniciar o GIT - git init
* Iniciar o versionamento - git add
* Criar um commit - git commit



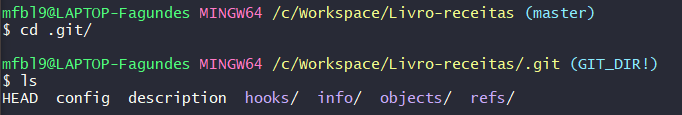
* Criação e visualização da novo diretório Livro-receitas



* Acesso ao novo diretório, criação do repositório do Git



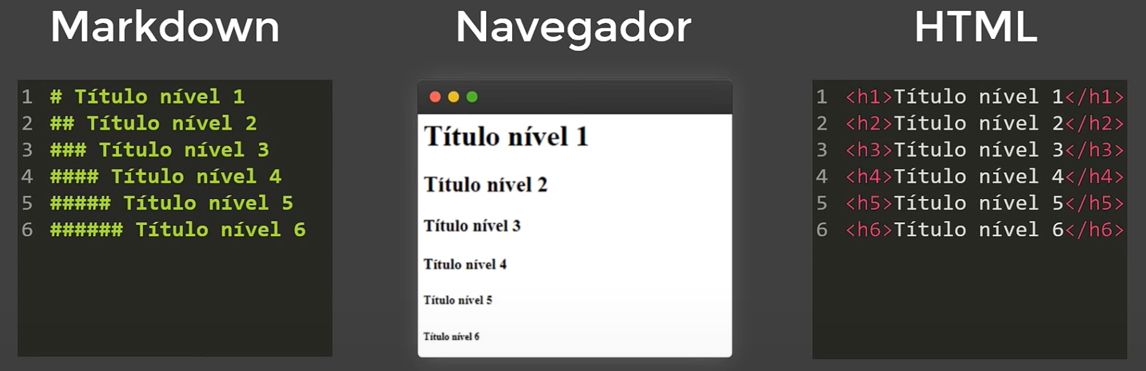
* Quando listamos o diretório Livro-receitas o repositório do Gitnão é mostrado por se tratar de uma pasta oculta
* A flag -a mostra arquivos ocultos



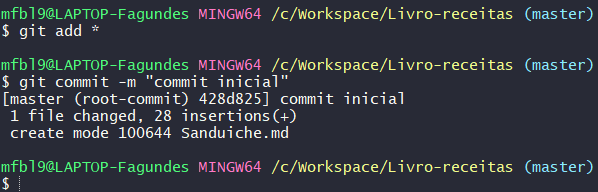
* Acesso a pasta do Git e visualização das informações presentes



* Configuração do email e no nome do usuário do Git



* Adicionando um arquivo



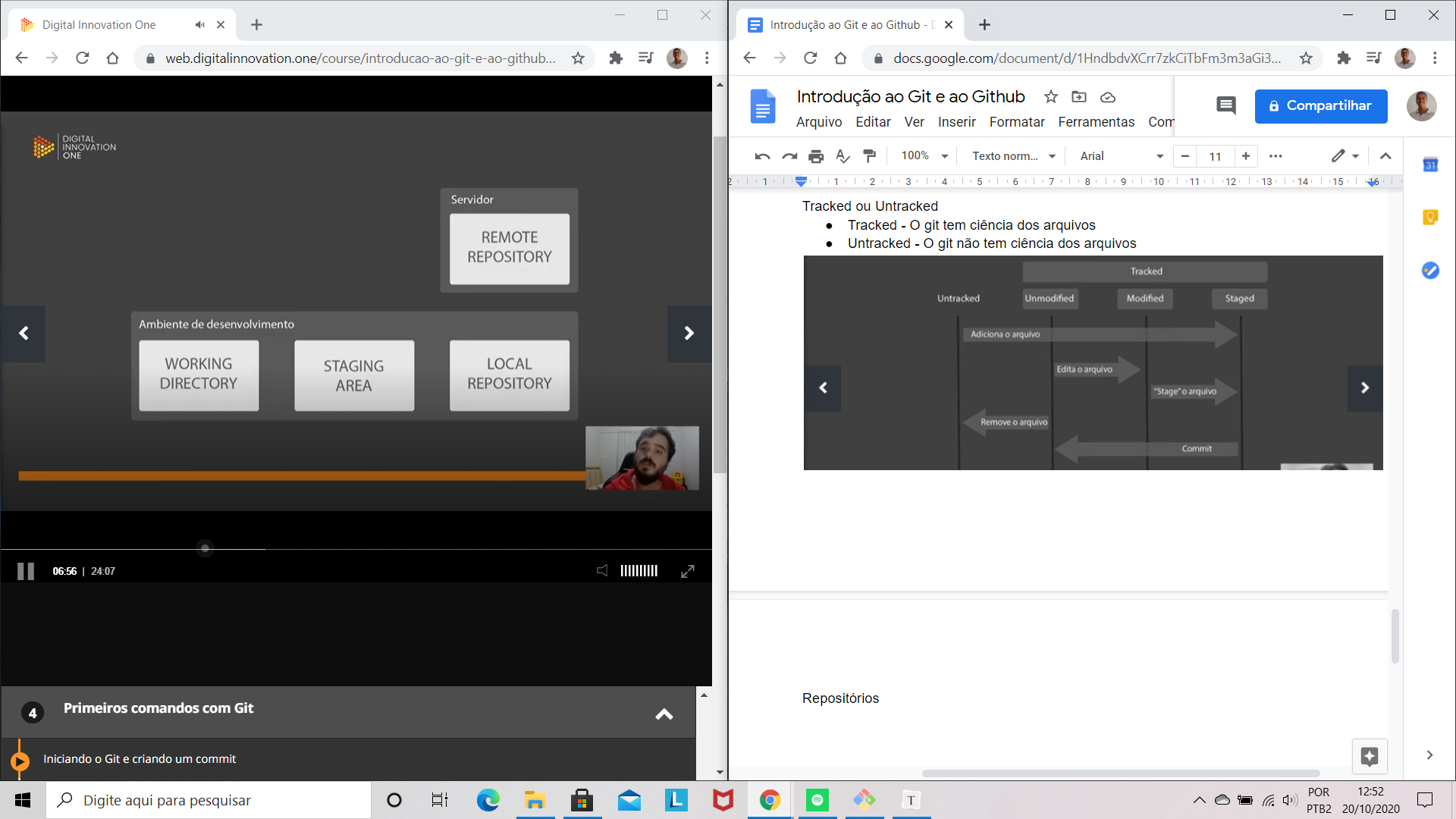
* -m (flag do git commit)
* 428d825 - Primeiros dígitos do SHA1
* git add \* adiciona todos os arquivos

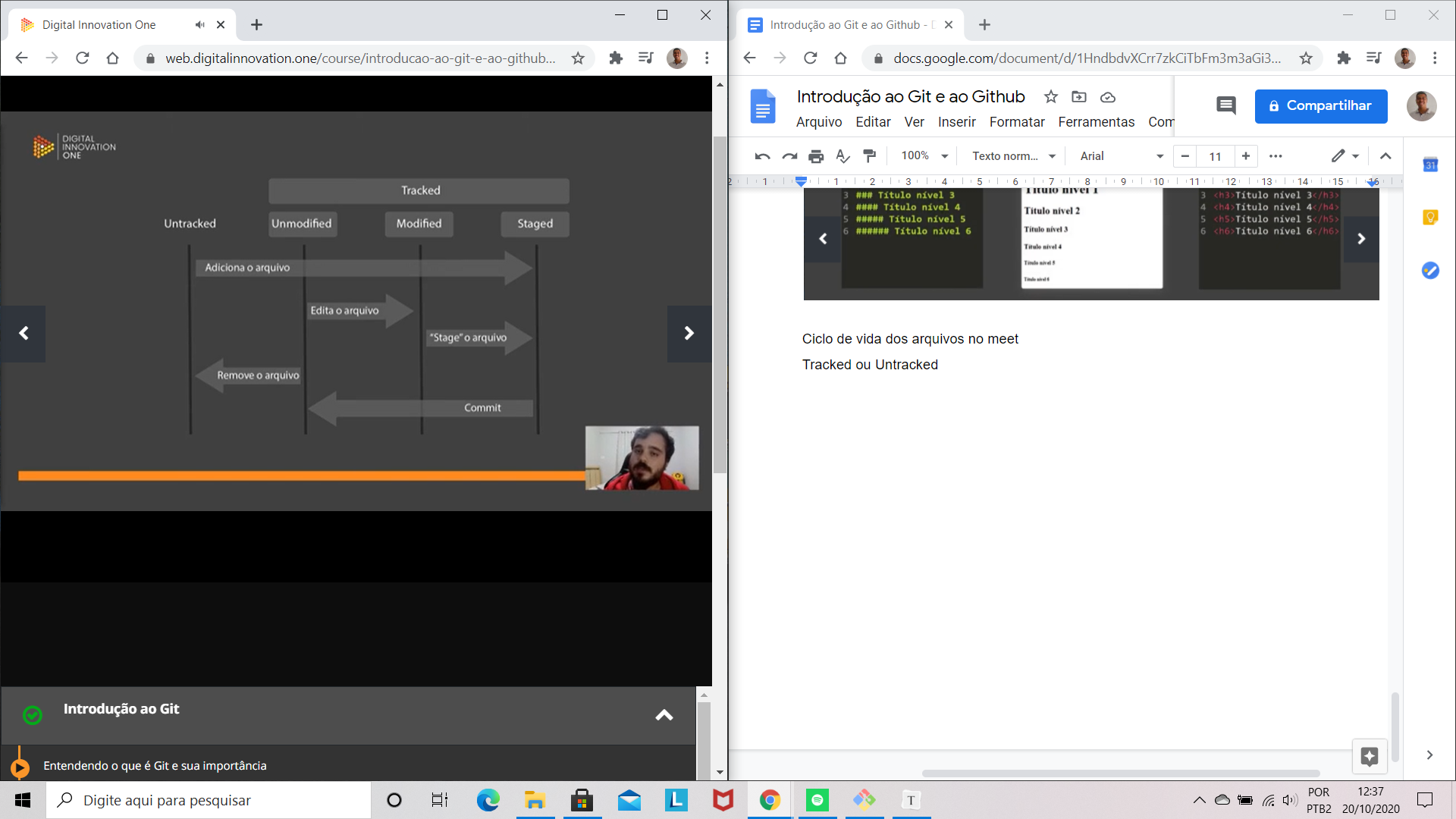
# Ciclo de vida dos arquivos no meet

Tracked ou Untracked

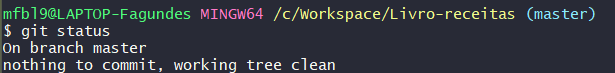
* Tracked - O git tem ciência dos arquivos
* Untracked - O git não tem ciência dos arquivos

Repositórios

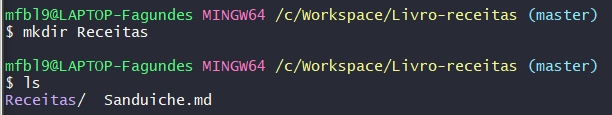




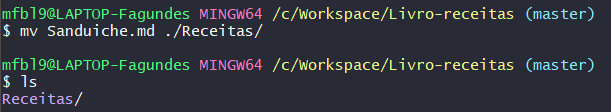
* Unmodified - Arquivo não modificado
* Modified - Arquivo modificado
* Staged - Arquivos em preparação para outro tipo de agrupamento, o commit



* git status
* Visualização do status do diretório



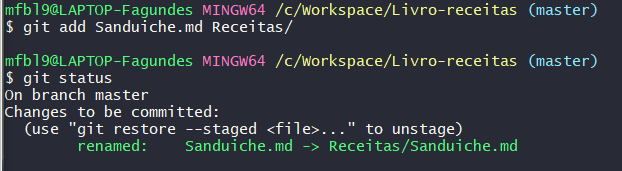
* Criação do diretório Receitas



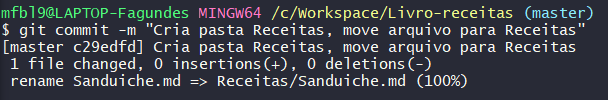
* Comando mv para mover o arquivo Sanduiche.md para a pasta Receitas

# 

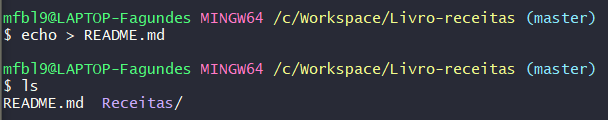
* Após criar a nova pasta e mover o arquivo para ela, quando executamos o comando git status recebemos as mensagens de exclusão do arquivo Sanduiche.md e que a Pasta Receitas não está restreada



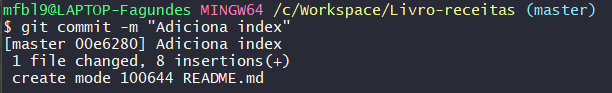
* Seguindo a sugestão do Git, usamos o comando git commit Sanduiche.md Receitas/ e dessa forma mudamos o status dos arquivos para staged e estão prontos para serem comitados.



* Dessa forma realizamos o commit com as alterações realizadas

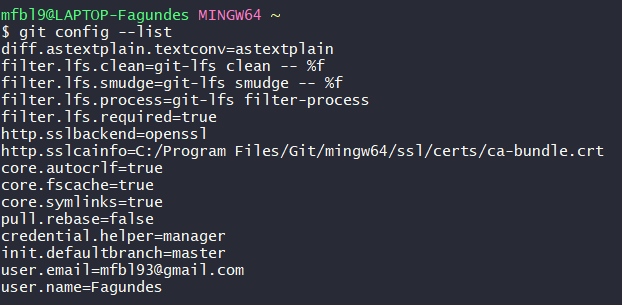


* Criação do arquivo README.MD



* Commit da adição do arquivo Index
* comando -m indica a adição de mensagem ao commit

# Introdução ao GitHub



* git config --list
  + O comando usado para listar as configurações do Git



* git config --global -unset (atributo)
  + Comando para limpar o atributo nas configurações



* Comando para setar um atributo nas configurações



* git remote add origin
  + O comando utilizado para apontar um repositório local para um repositório remoto

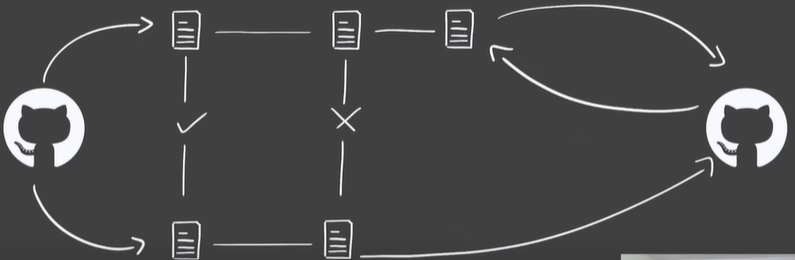
O GitHub lê arquivos Markdown

Quando você gera commits no seu repositório local com email e nickname diferentes dos utilizados no GitHub

* Os commits aparecerão com email e nickname diferentes no GitHub

O GitHub também é uma rede social

# Resolvendo conflitos



* O que é um conflito?
  + Quando duas pessoas editam a mesma linha de código (conflito de merge)

Após corrigir o conflito em seu repositório local é necessário

* Criar um novo commit e enviá-lo (git push) para o GitHub

O Git não resolve todos os conflitos de merge automaticamente

O comando git clone

* Faz um clone do repositório remoto em seu repositório local

Quando você tenta empurrar (push) o código para o GitHub e o seu commit não representa o estado mais atual do repositório o git irá:

* Pedir para você fazer um pull e depois um push