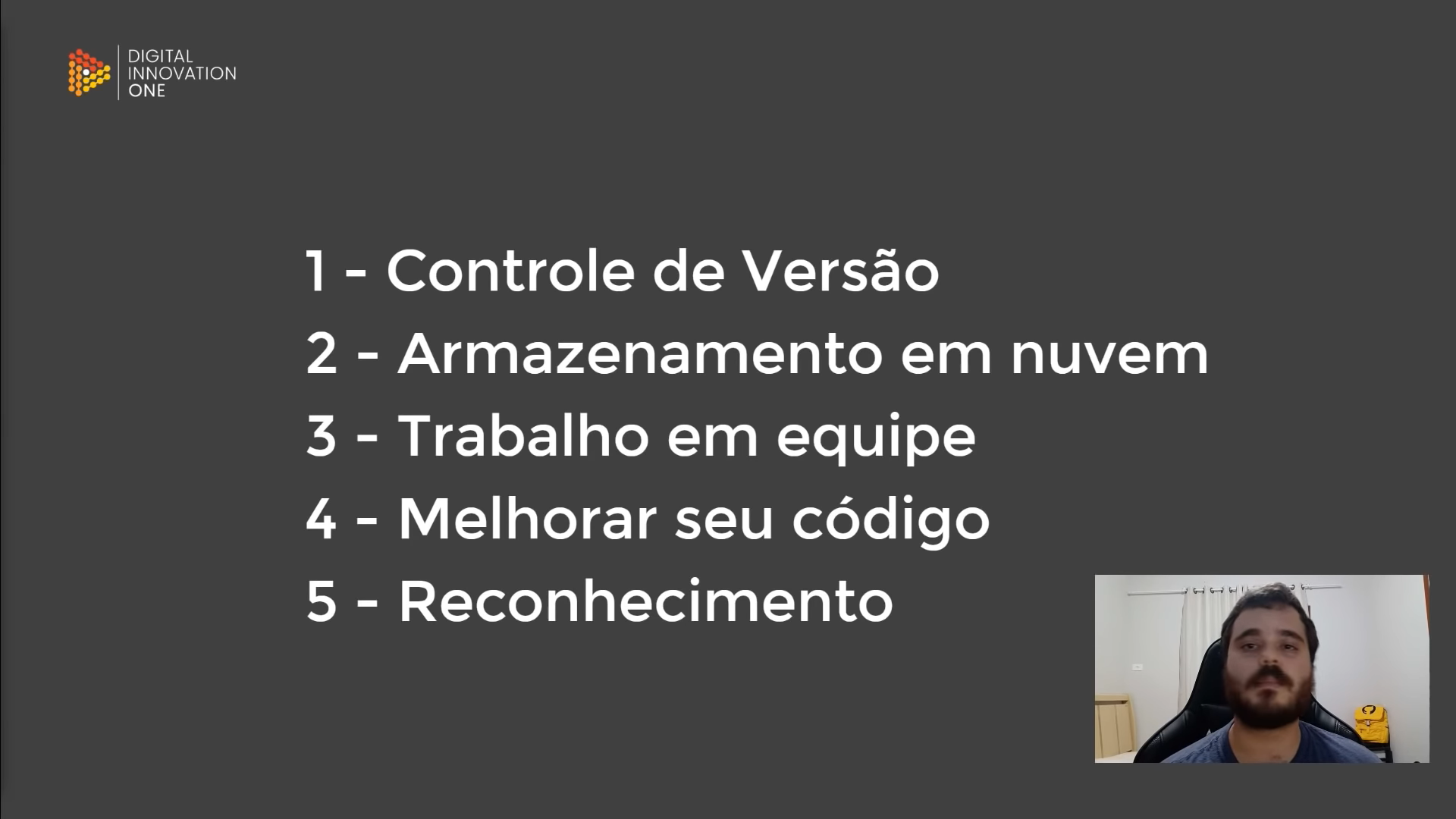
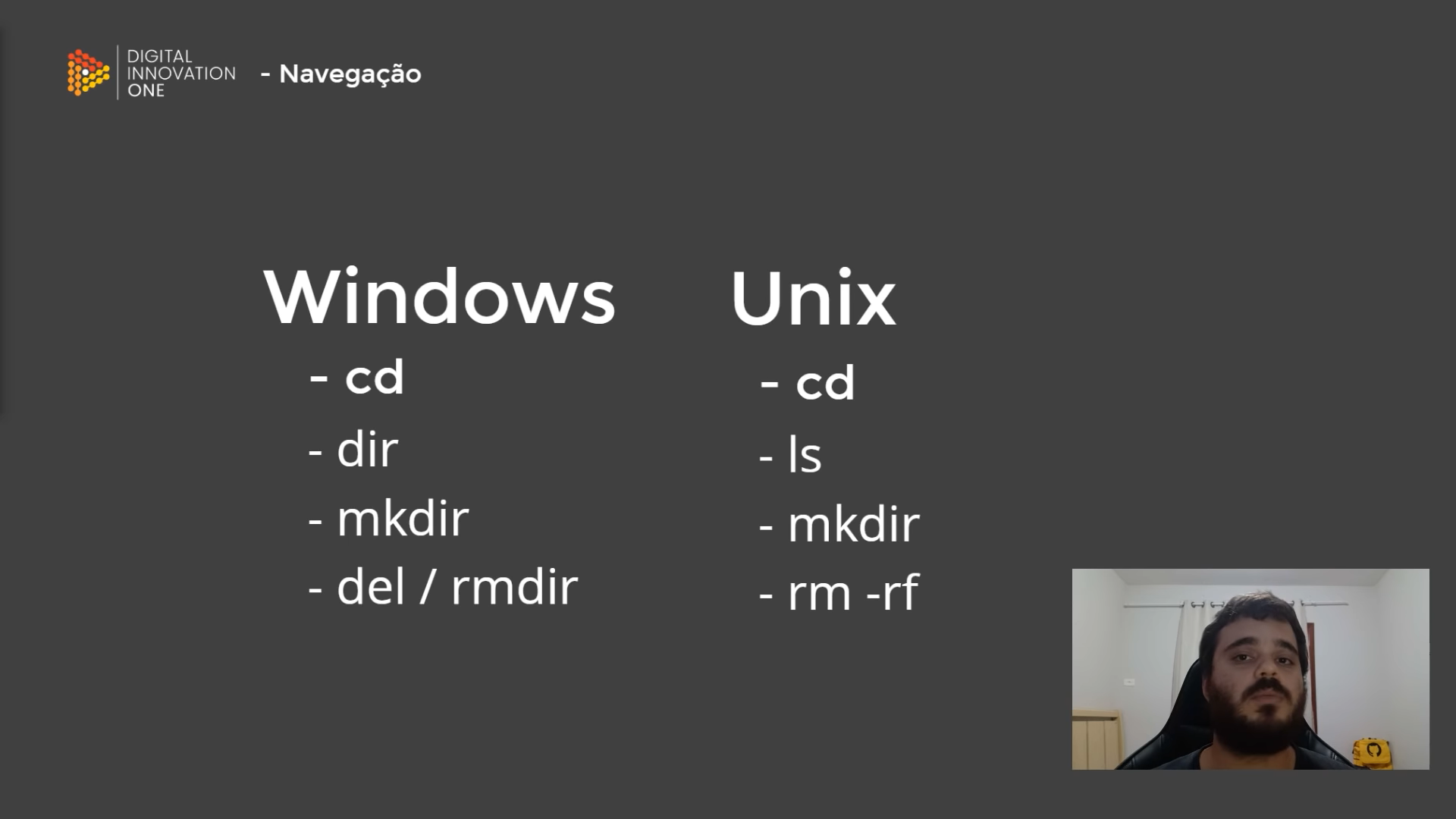
**Git e Git Hub**



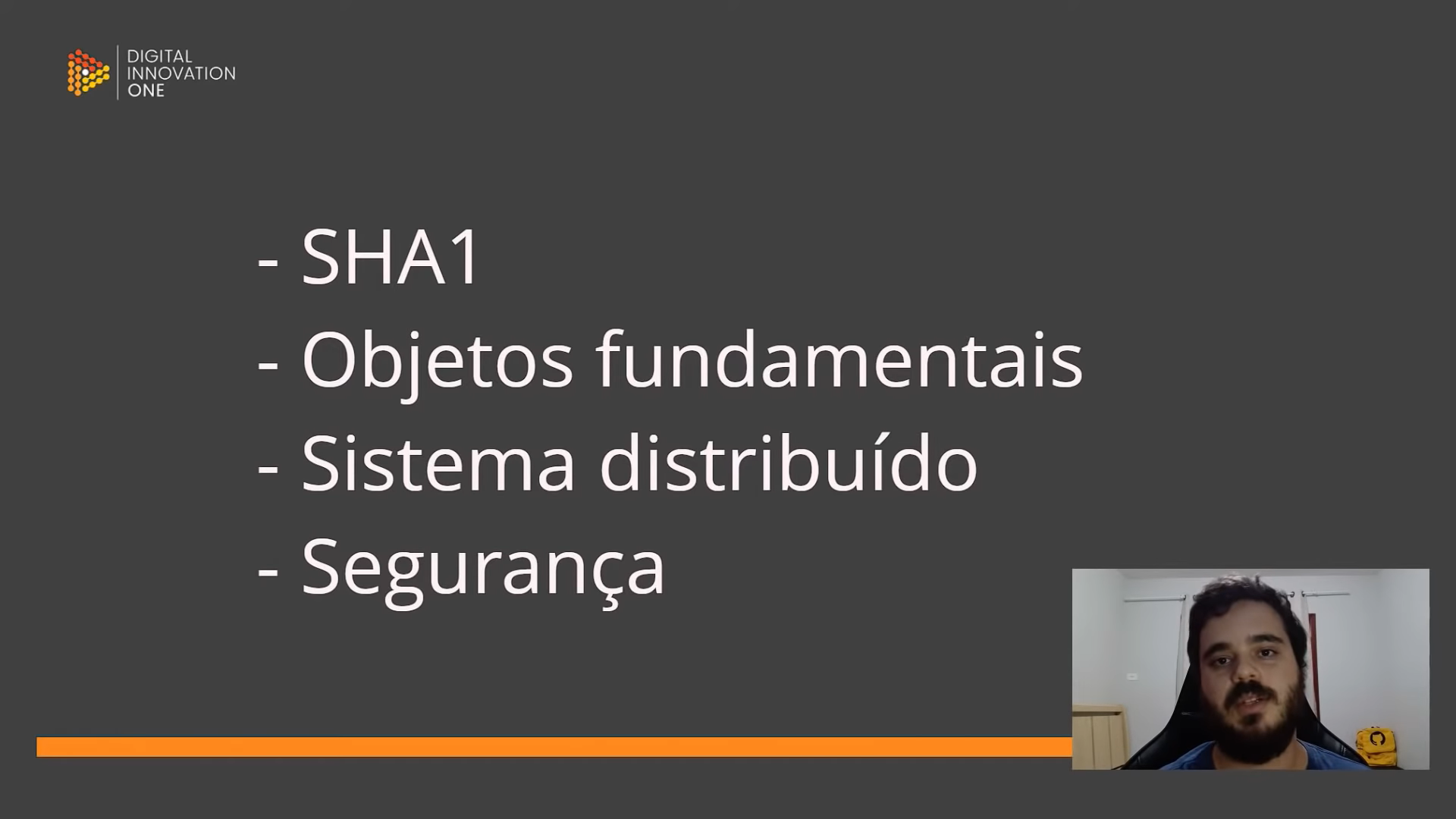
**CLI** = Comand Line Interface Não tem interface gráfica ( a intereção é por linhas de comando )

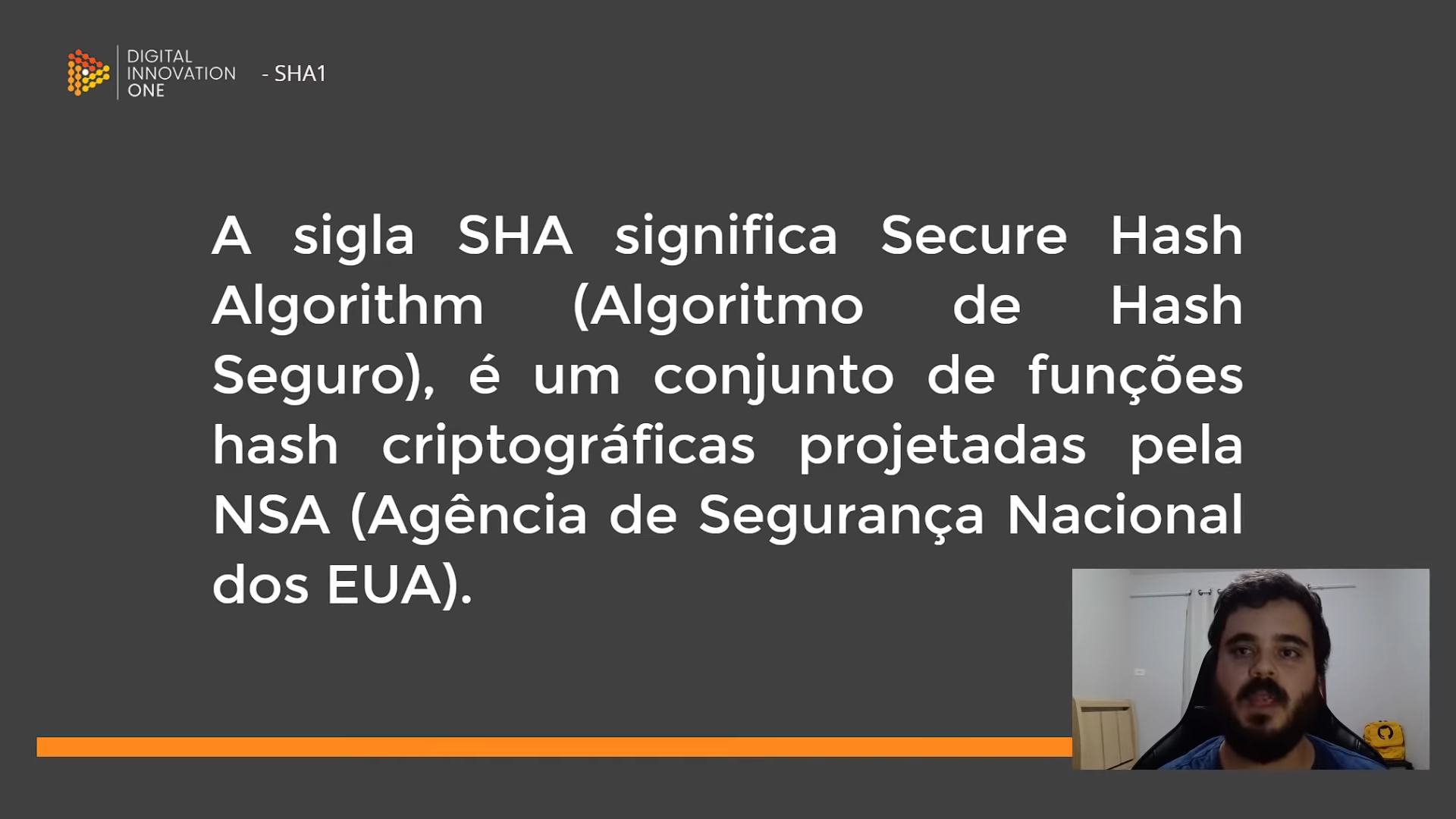
Alguns comandos são diferentes nos S.O

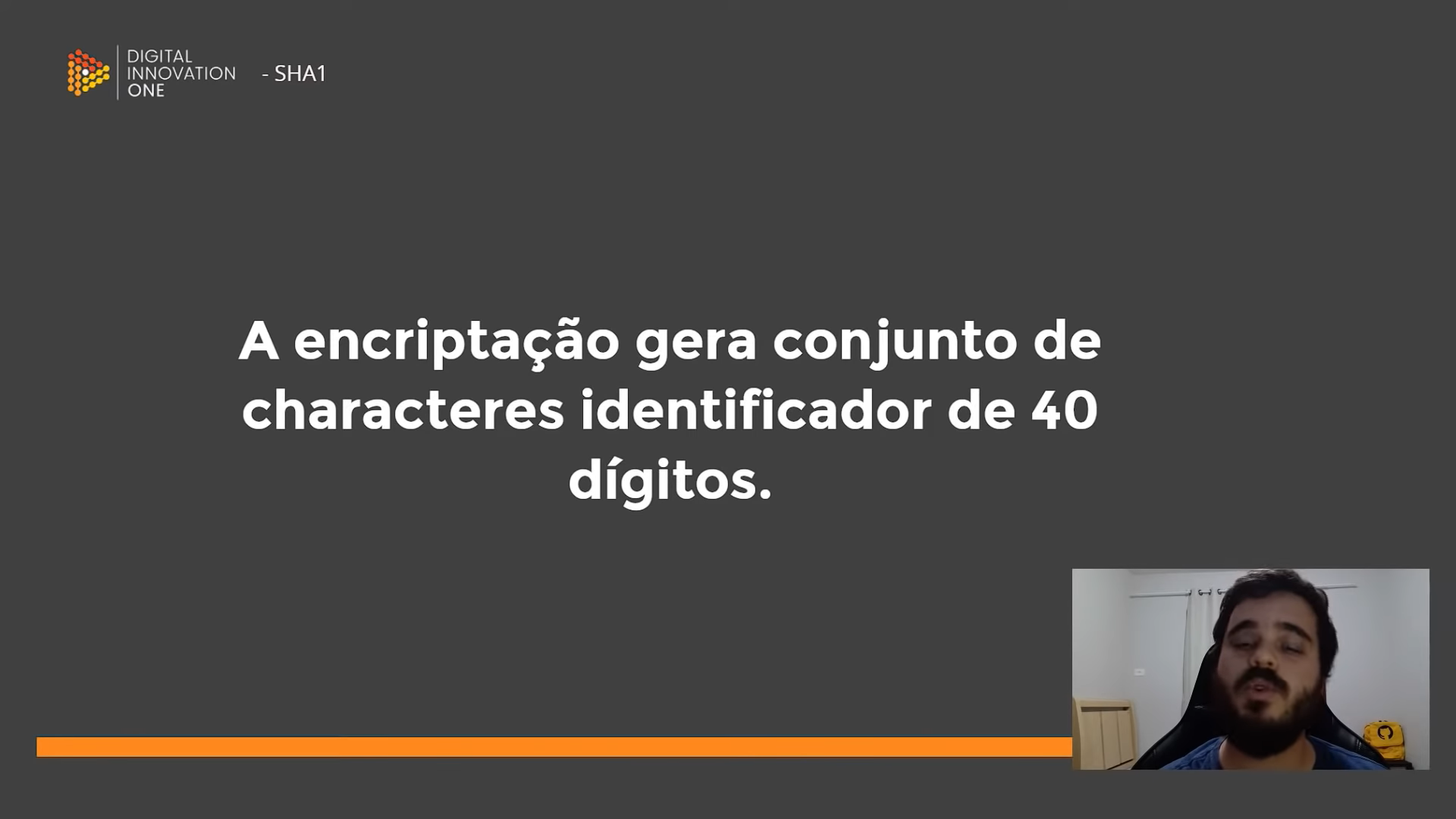


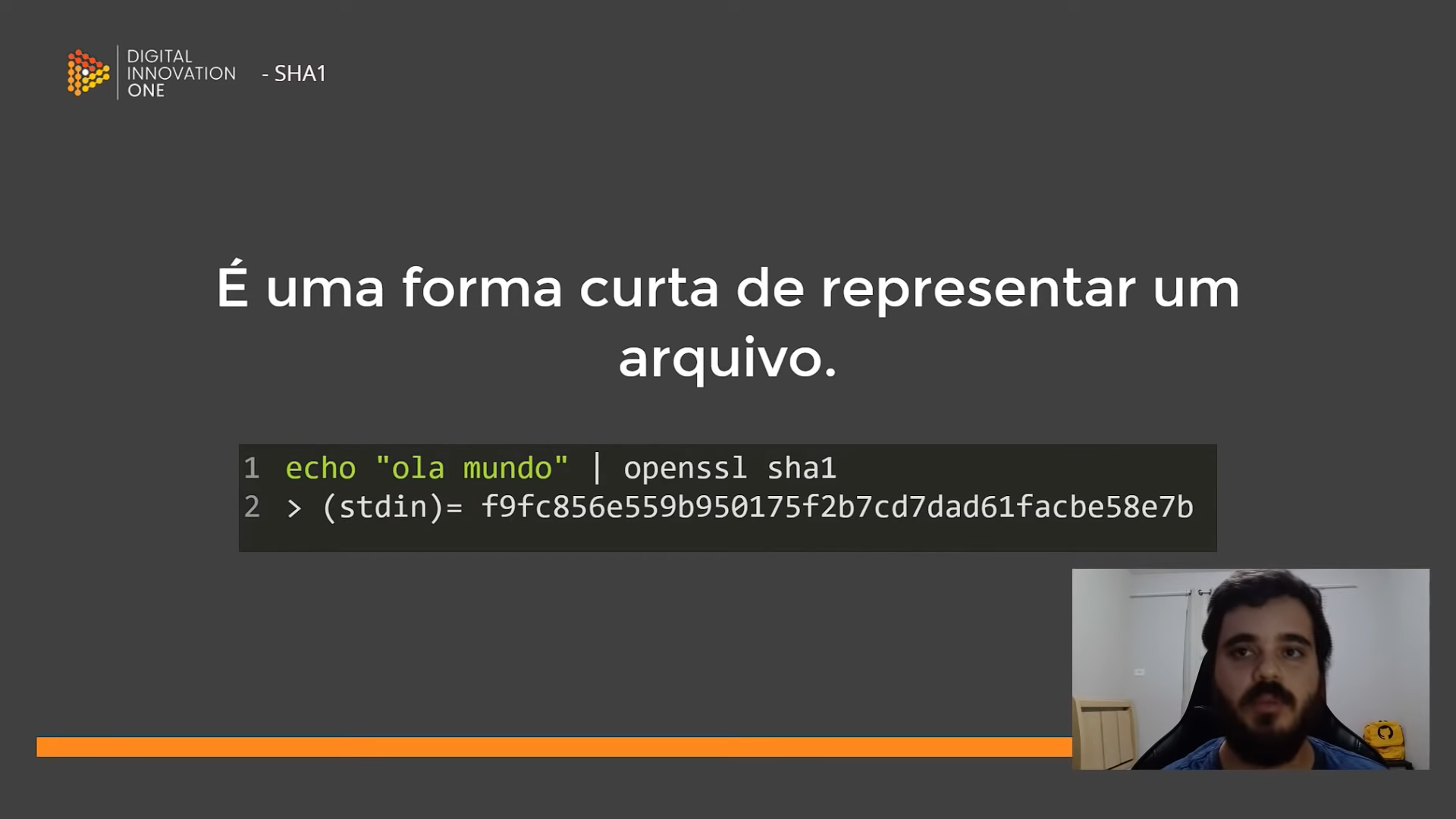
**Comandos mais usados no Dos ou terminal Linux:**

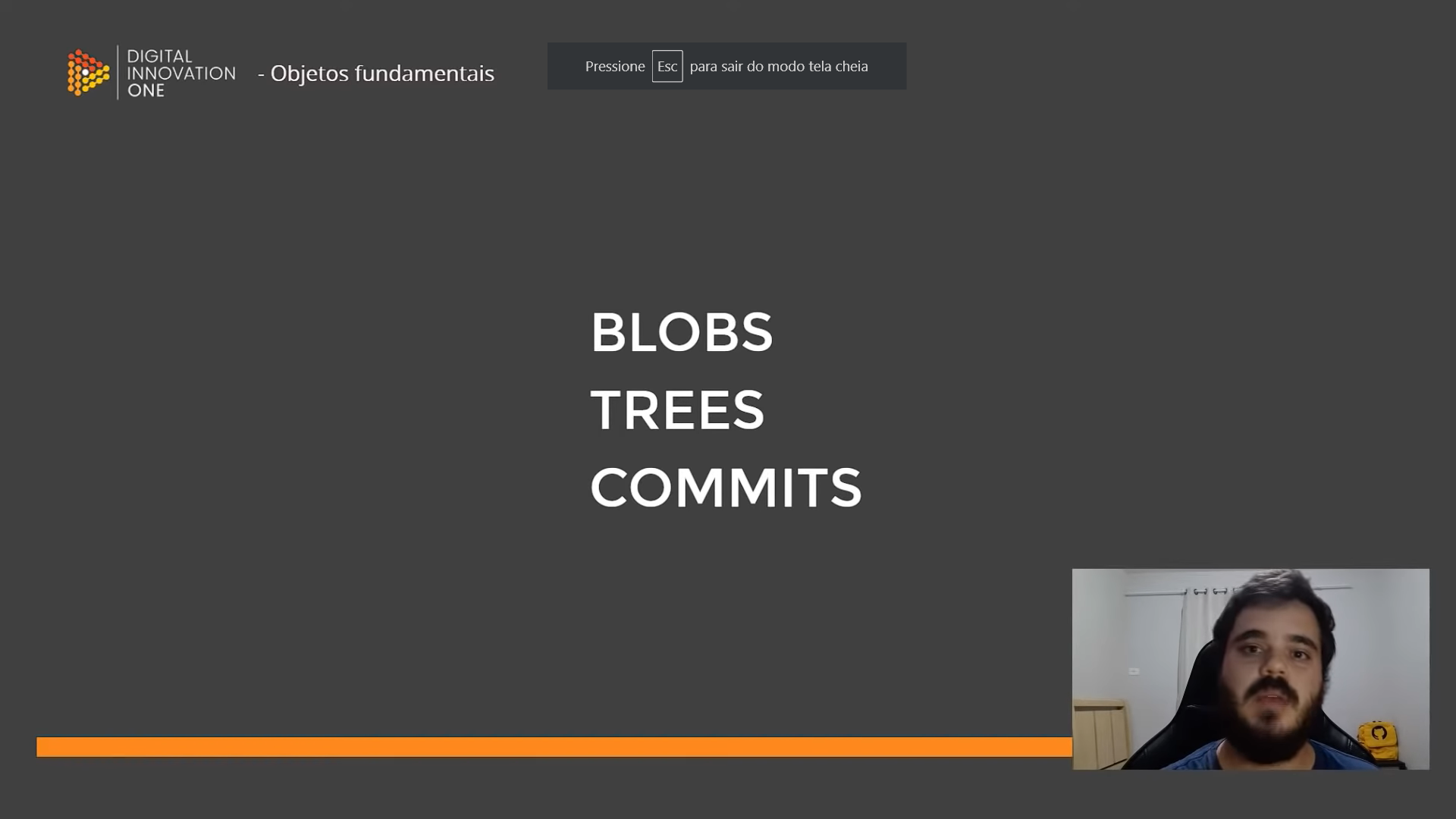
* **dir** = lista as pastas (no Windows) **ls** = lista pastas no (Linux)
* **cd** = para navegar até um local especifico no sistema ex( cd/ vai para o diretório raiz c:)
* **cd..** = retrocede 1 pasta ou seja se estiver dentro de uma pasta ele volta ao diretório raiz
* **cls** = limpa a tela no Windows **clear** = limpa a tela no Linux
* **cd + table** = autocompleta a pasta com a letra inicial que vc digitou ex: se digitar w ele auto completa Windows se existir uma pasta com esse nome.
* **mkdir =** cria uma pasta
* **echo =** cria um arquivo ex: (echo hello > hello.txt ) cria um arquivo chamado hello com extensão .txt que é um bloco de notas
* **del =** deleta um arquivo no Windows, mas não deleta a pasta
* a tecla para cima lista o histórico de comandos realizados no diretório anteriormente.
* **rmdir** = apaga a pasta e todos os arquivos que estão dentro dela ex: C:\>rmdir workspace /s /q **no Linux** utiliza o **rm -rf workspace /**

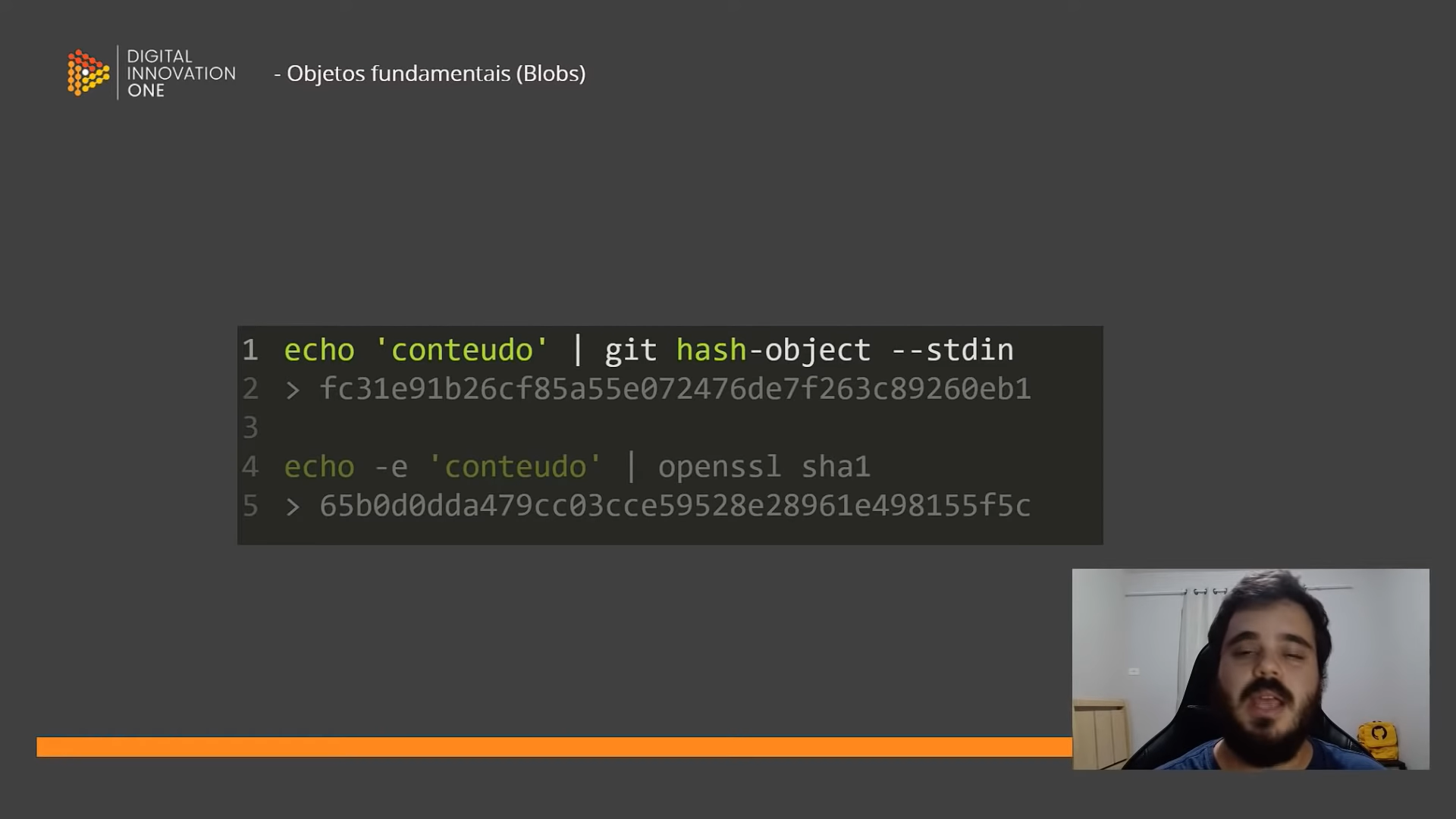












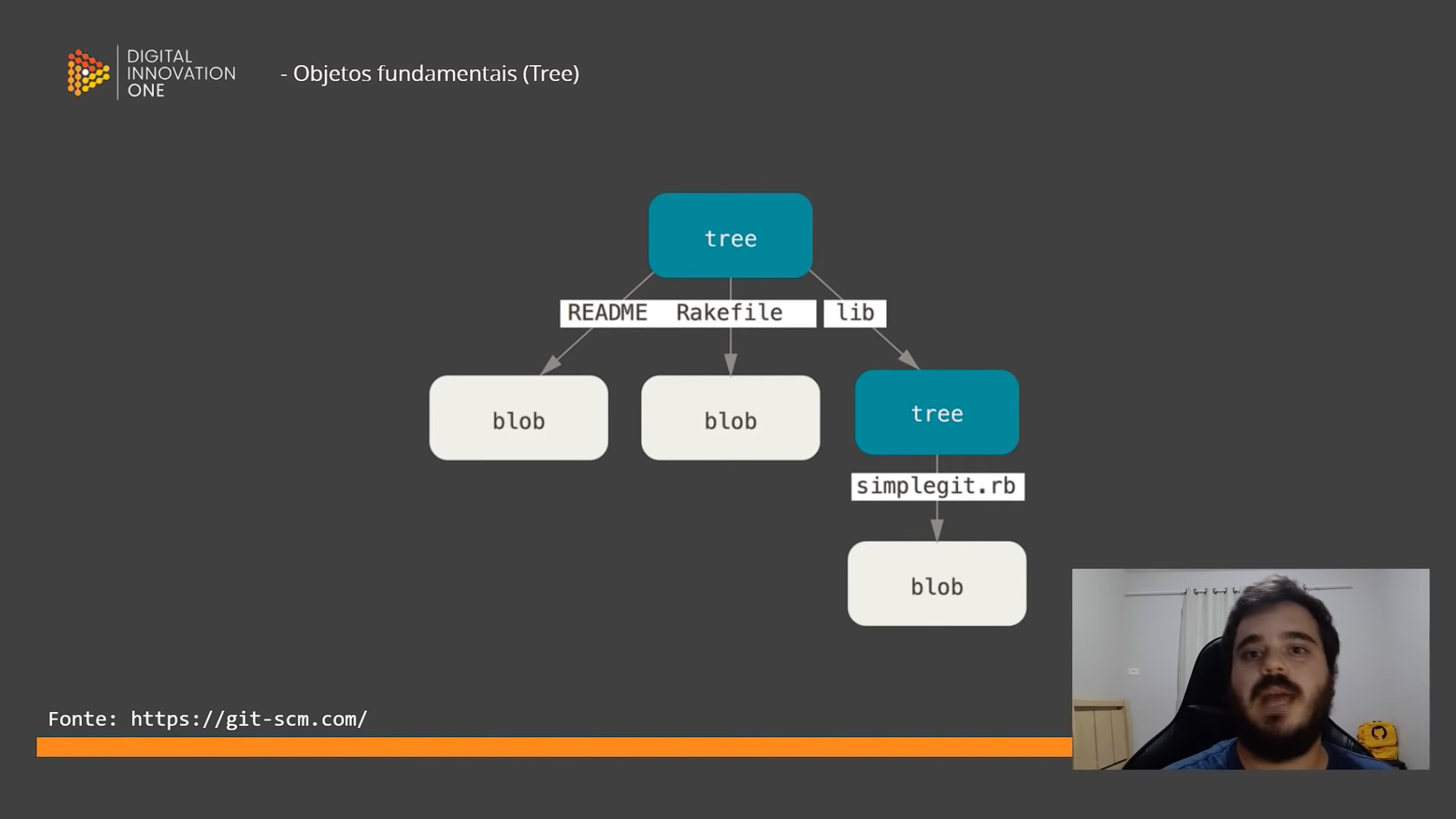
Tree (Arvore)

* Guarda o nome do arquivo
* Aponta para outro arquivo
* Aponta para outras arvores
* Possuem Sha1

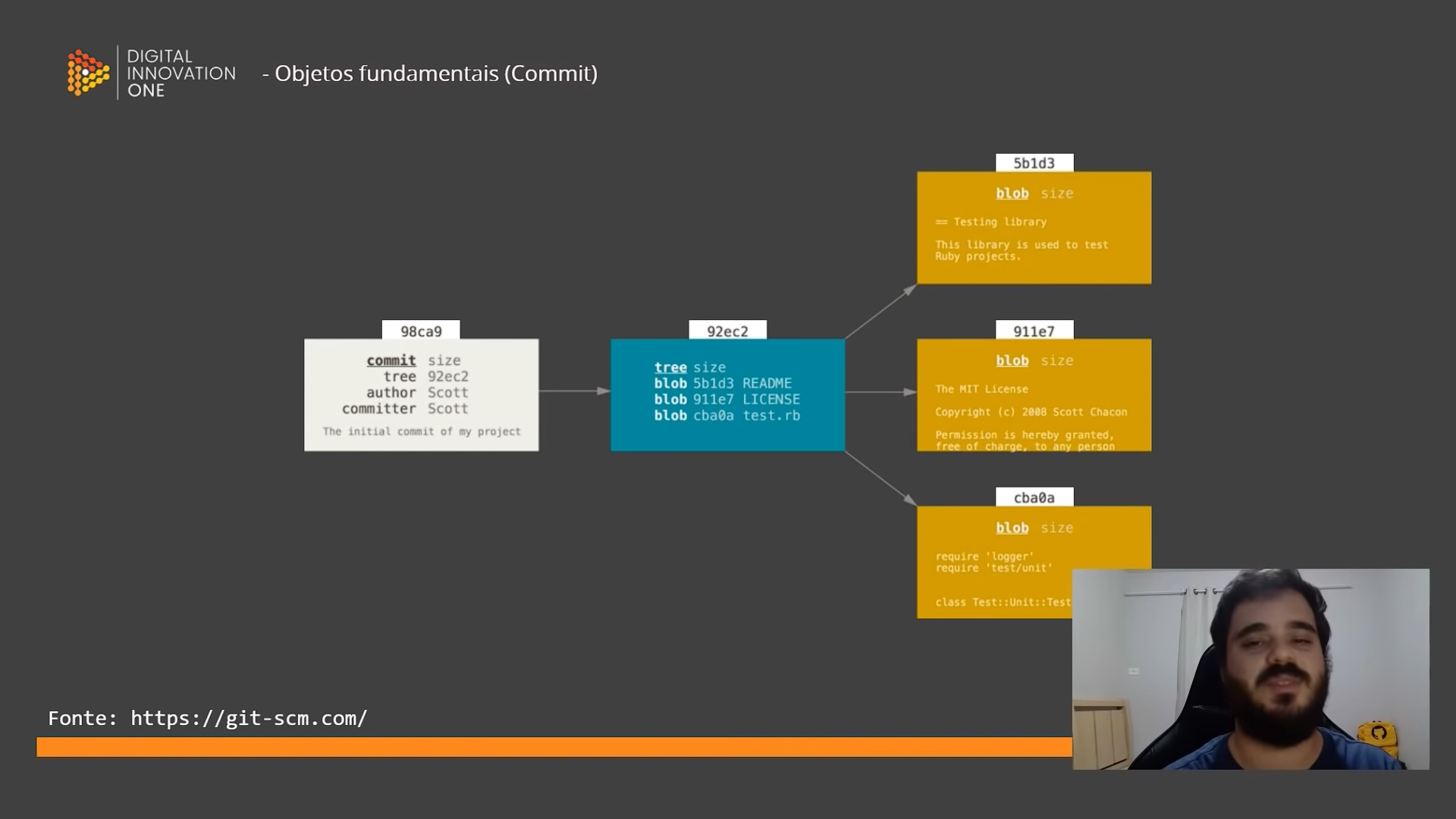
Blob (Bolhas)

* Só guarda o sha do arquivo

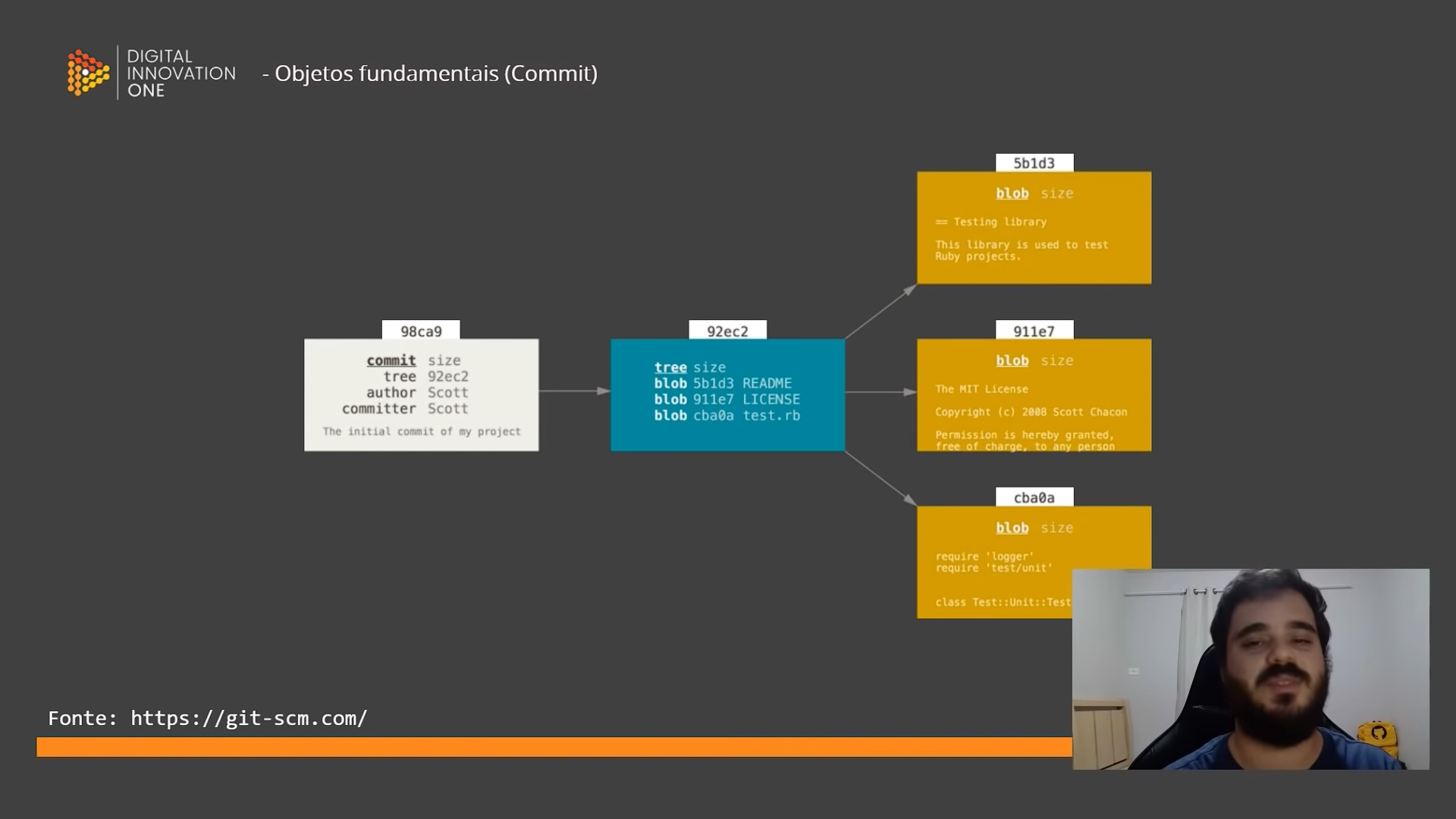
Comid











Chave SSH = conexão seguro entre 2 maquinas

Comnado para gerar a chave publica e privada dentro do gitbash:

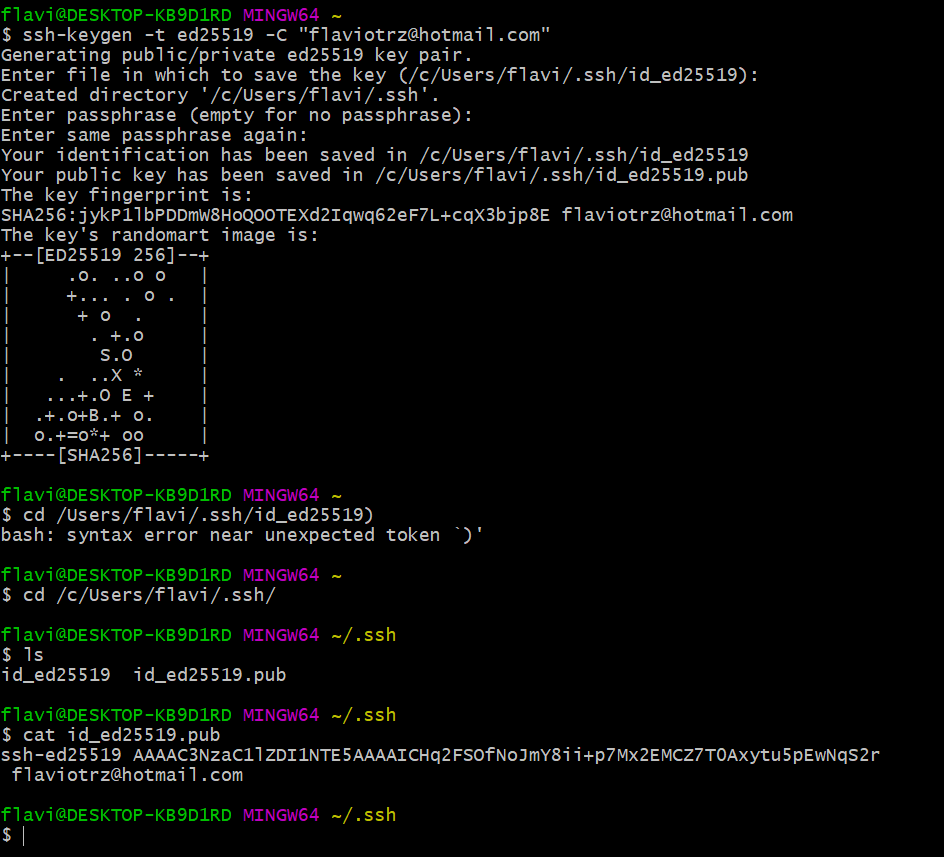
$ ssh-keygen -t ed25519 -C "flaviotrz@hotmail.com"

Acessar a pasta onde esta as chaves:

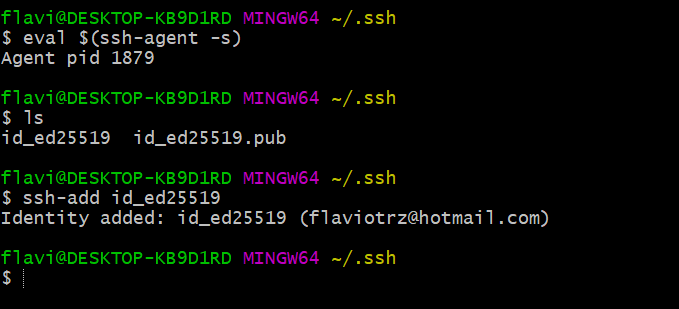
$ cd /c/Users/flavi/.ssh/

Lista os aquivos com um comando LS

PROCESSO DE CRIAÇÃO DA CHAVE SSH para o GITHUB



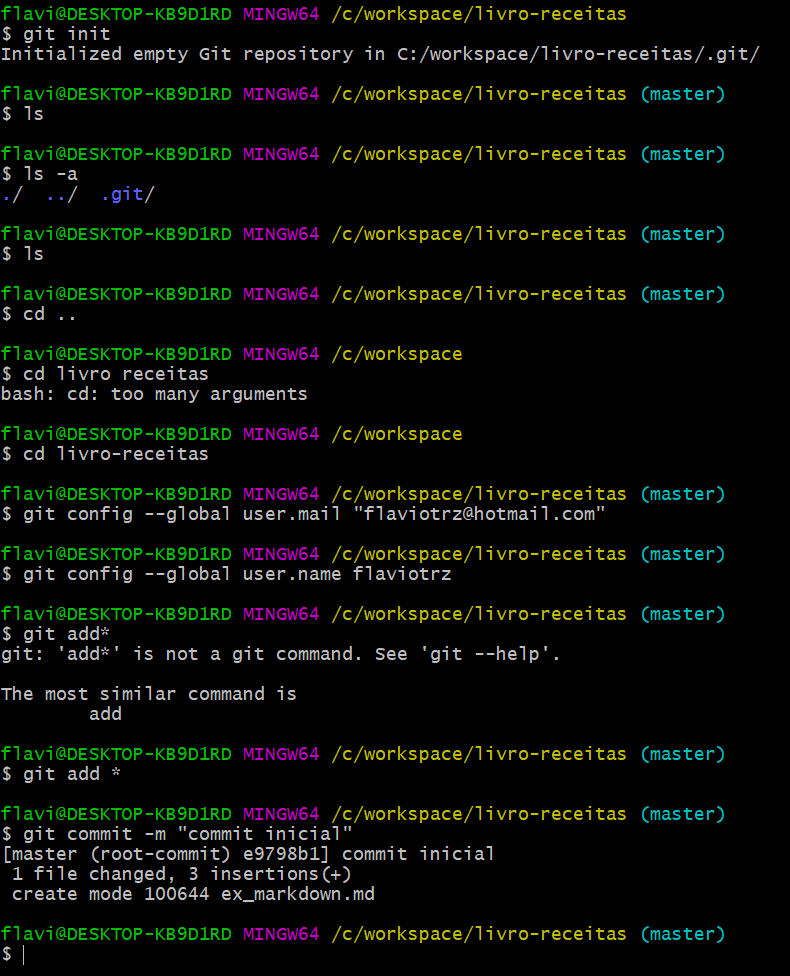
Adiciona a chave privada no agente

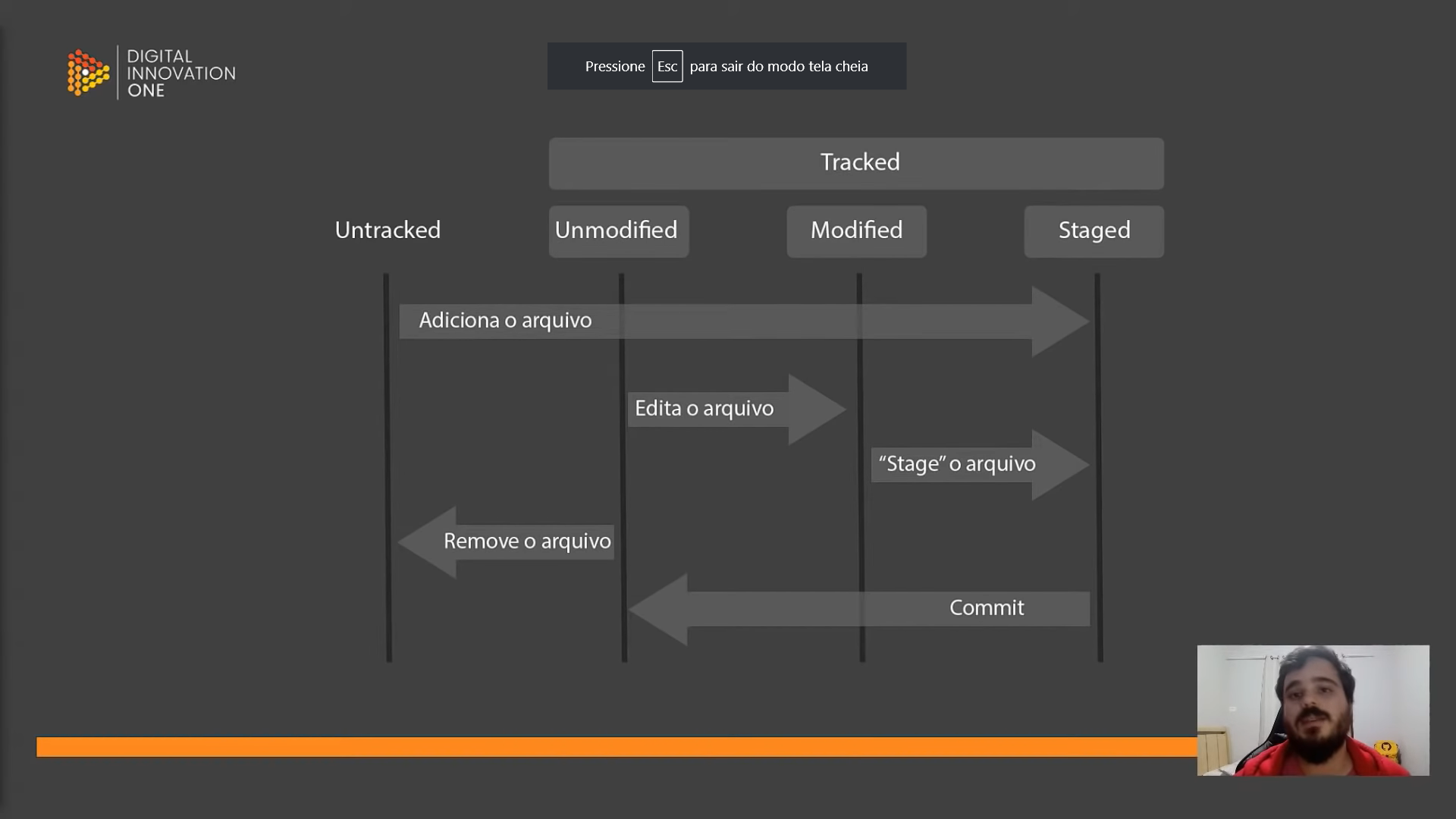




**git init** = dentro da pasta de arquivos com o git bash aberto executa esse comando para inicializar a sincronização dessa pasta com o repositório da nuvem (github)

**ls -a** = exibe pastas ocultas





Untracked = um arquivo que o git ainda não sabe da sua existência

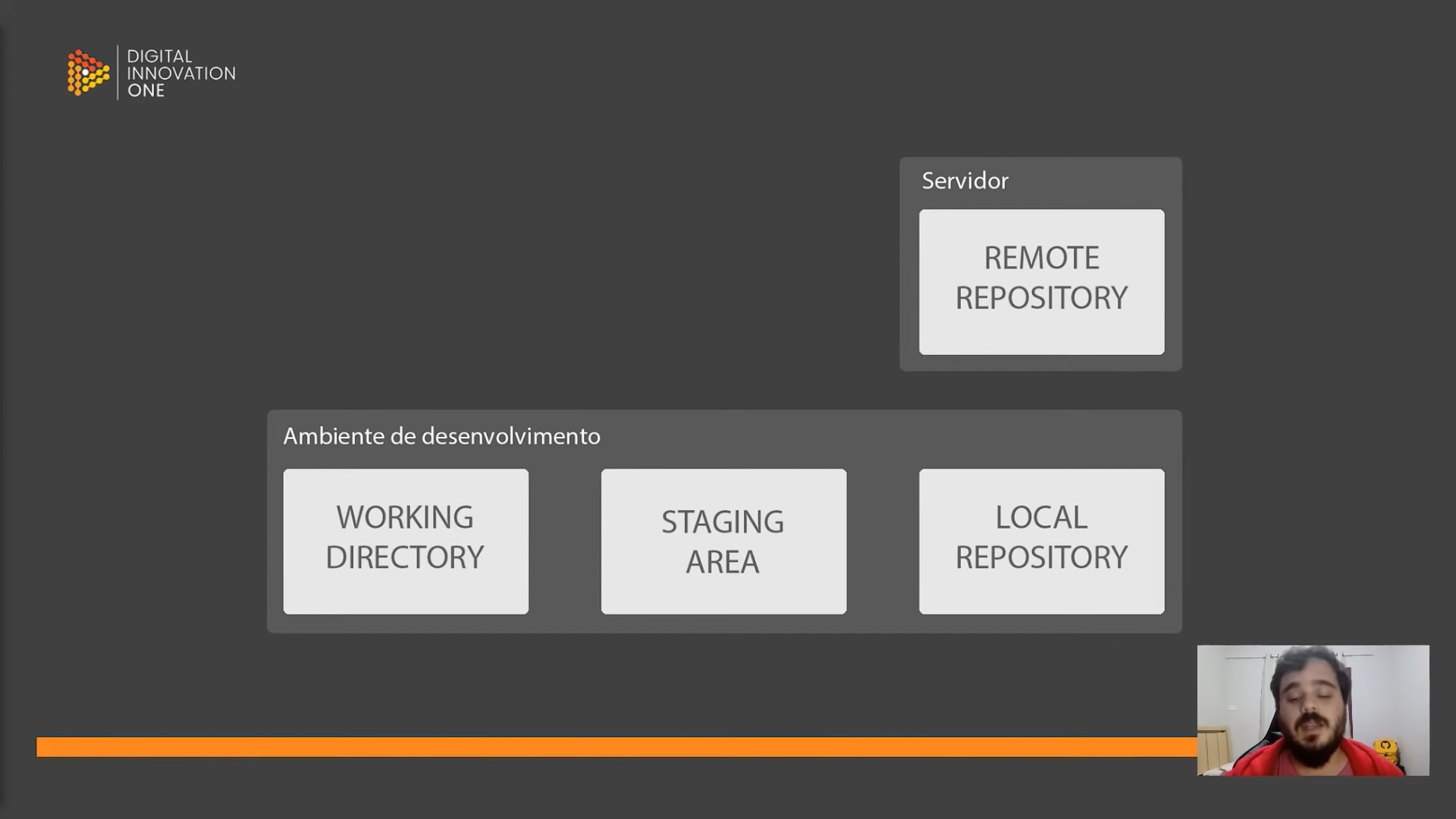
Ao iniciar o comando Add estamos fazendo com que o git tenha a ciência da existência desse arquivo.

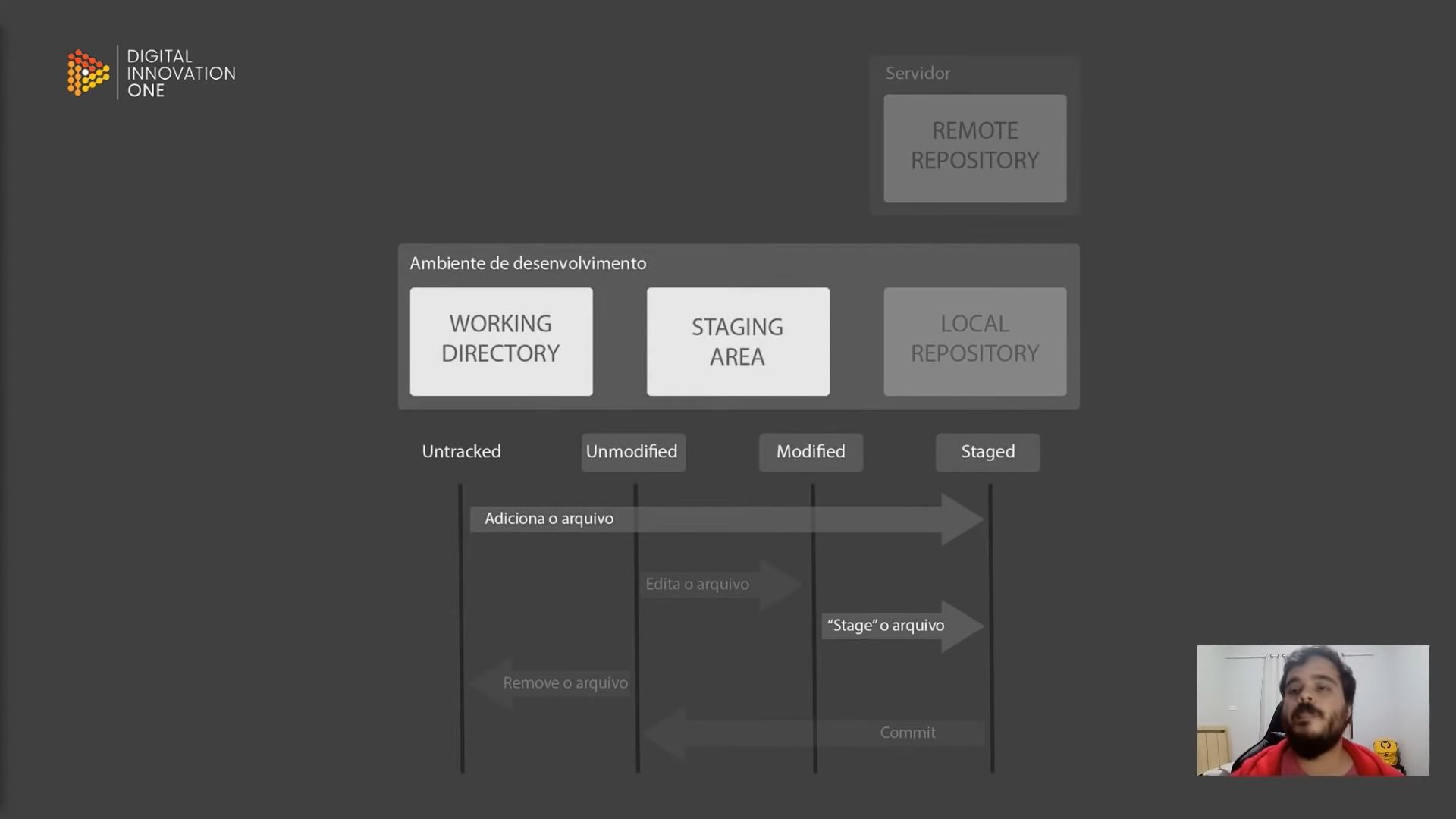
Unmodified = é um arquivo dentro do repositório do git que ainda não sofreu nenhuma modificação.

Modified = quando o arquivo é aberto e feito modificações.

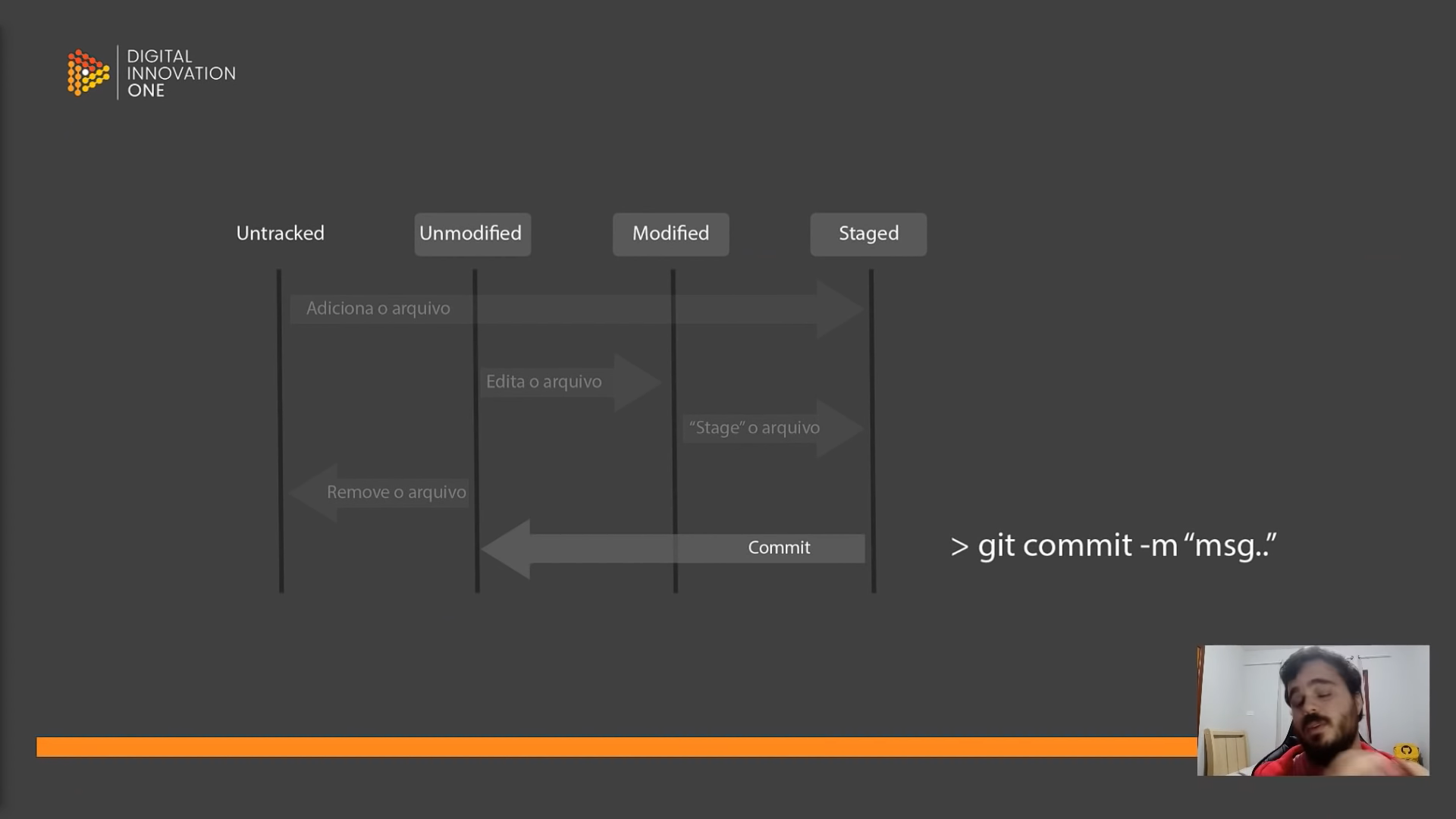
Staged = área onde o arquivo fica aguardando para executar outra ação

Commid = todo o arquivo é encapsulado com todas as modificações realizadas e está pronto para sair da área de estagio, depois ele retorna esse mesmo arquivo para unmodified de novo para que assim que outra alteração seja realizada ele gere um novo arquivo contendo essas ultimas modificações gerando assim um histórico de trabalho nesse arquivo.

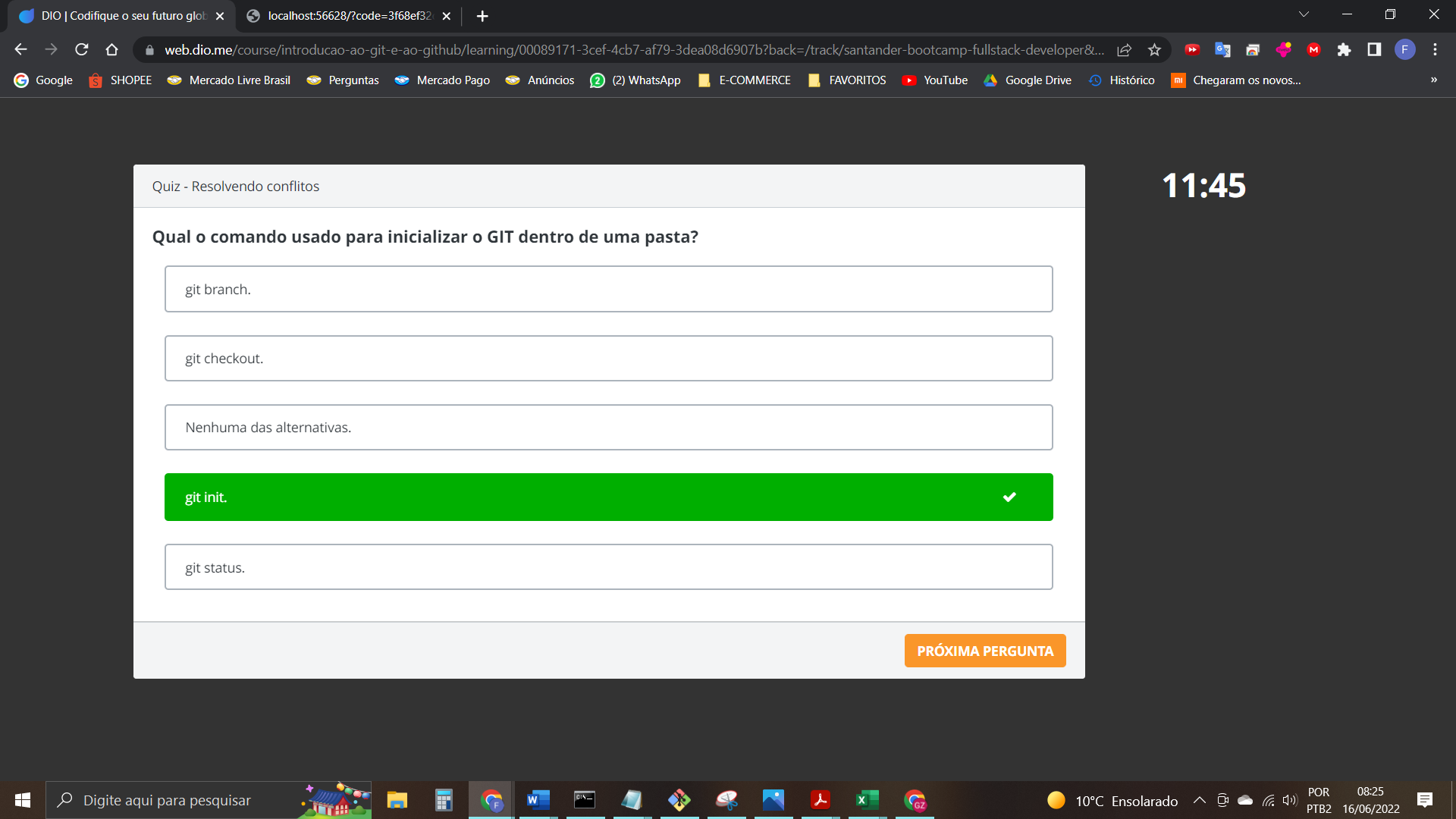


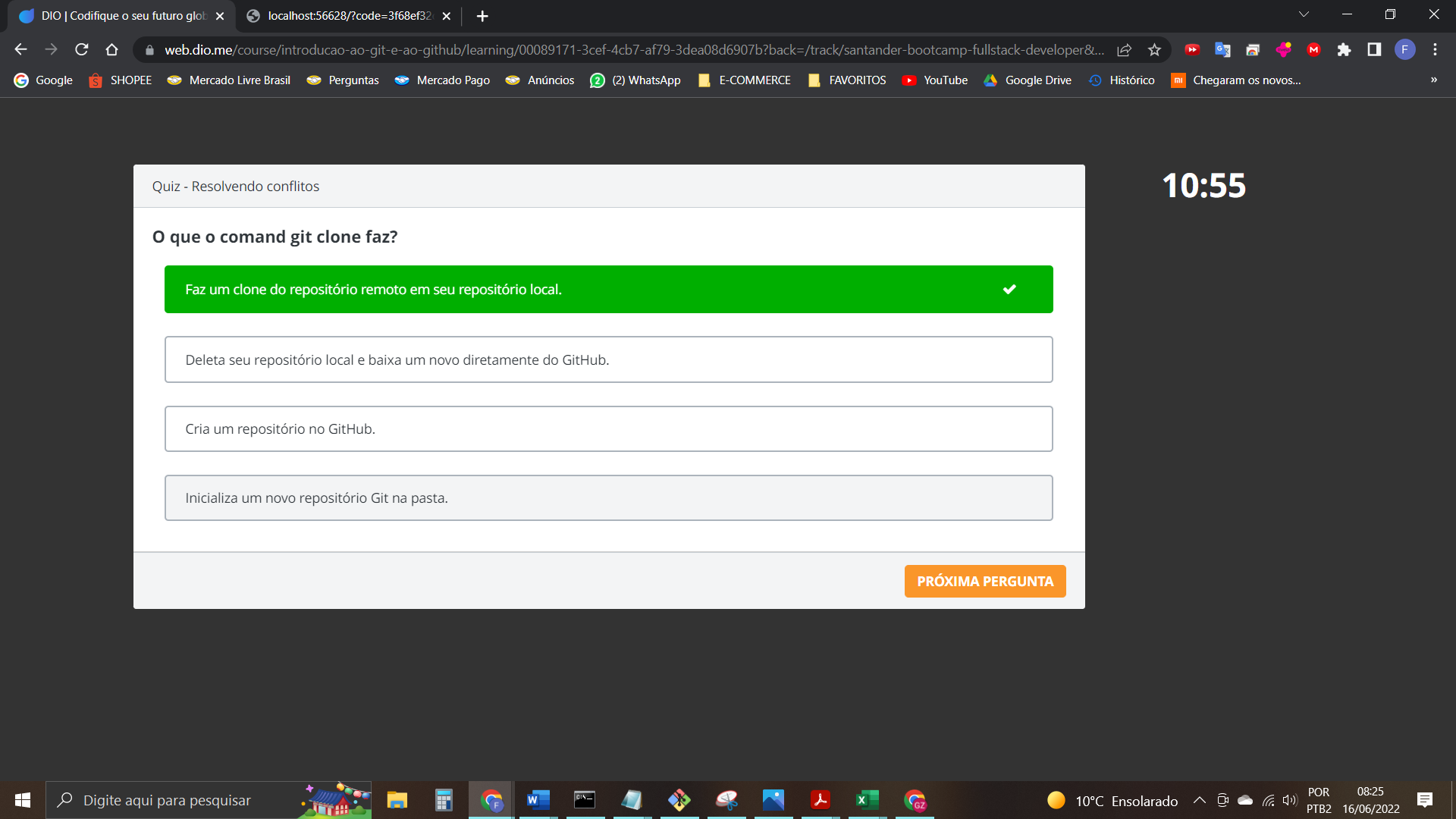


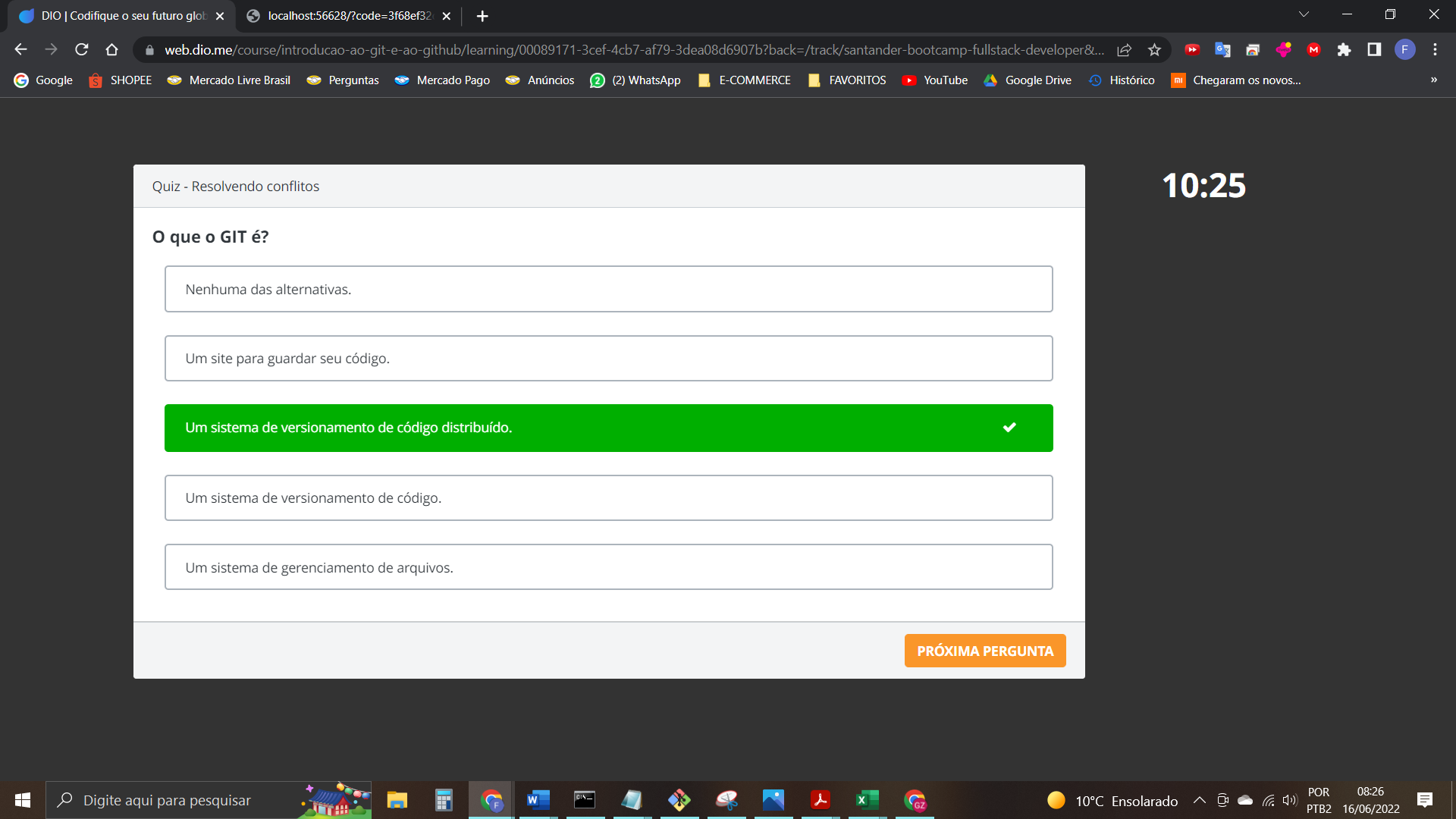


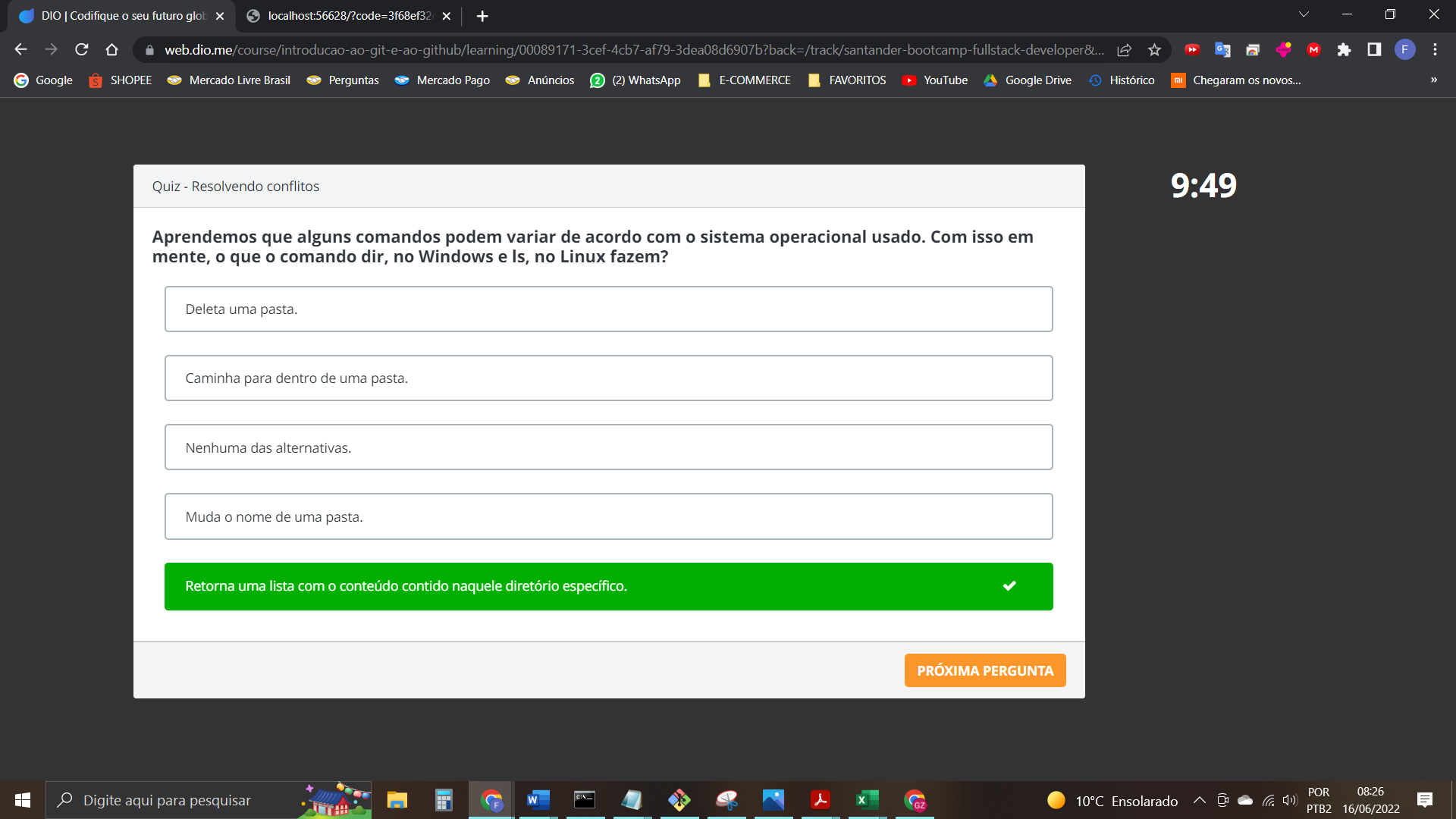


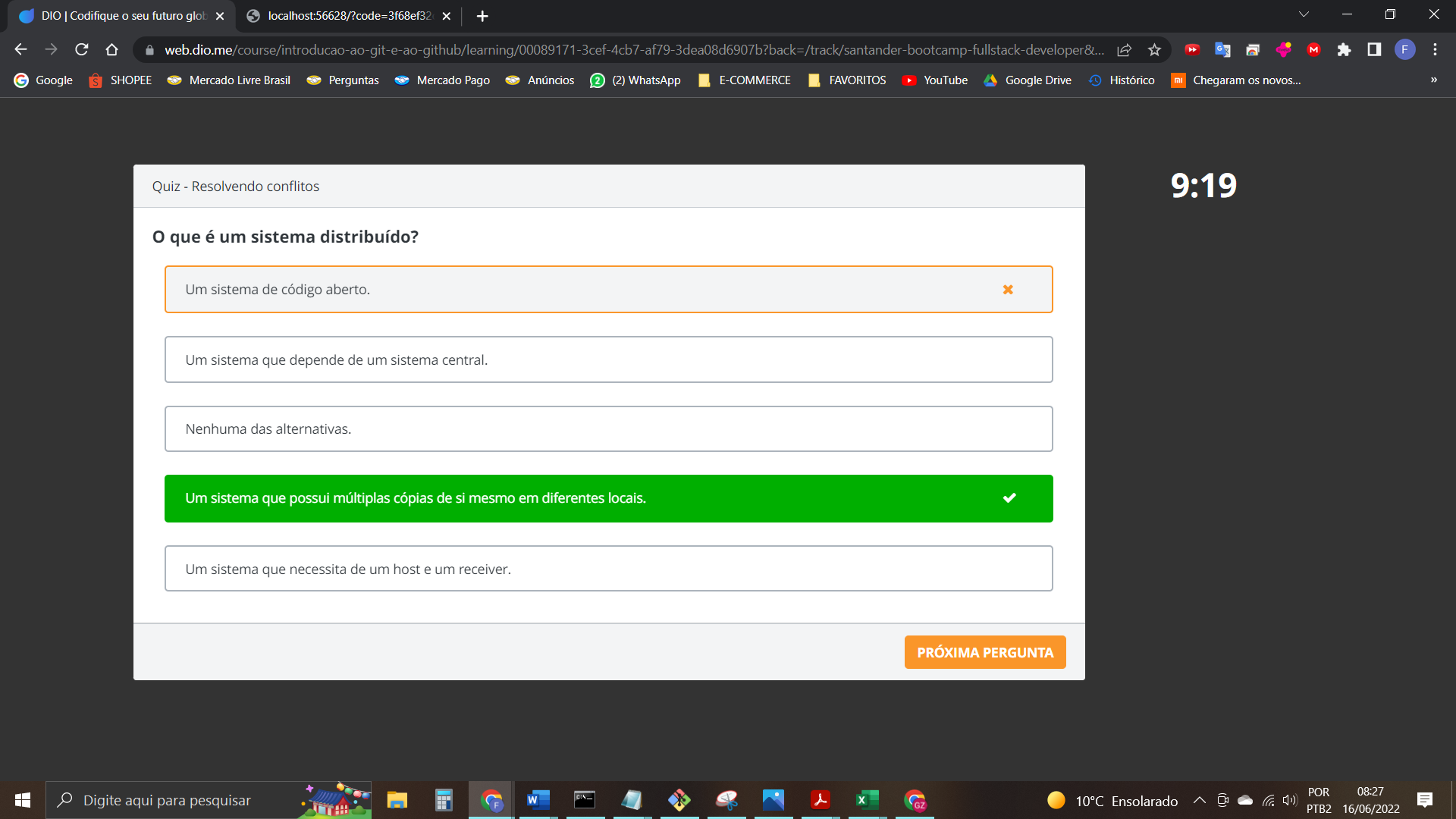
**git config –list** (para verificar as configurações do github)

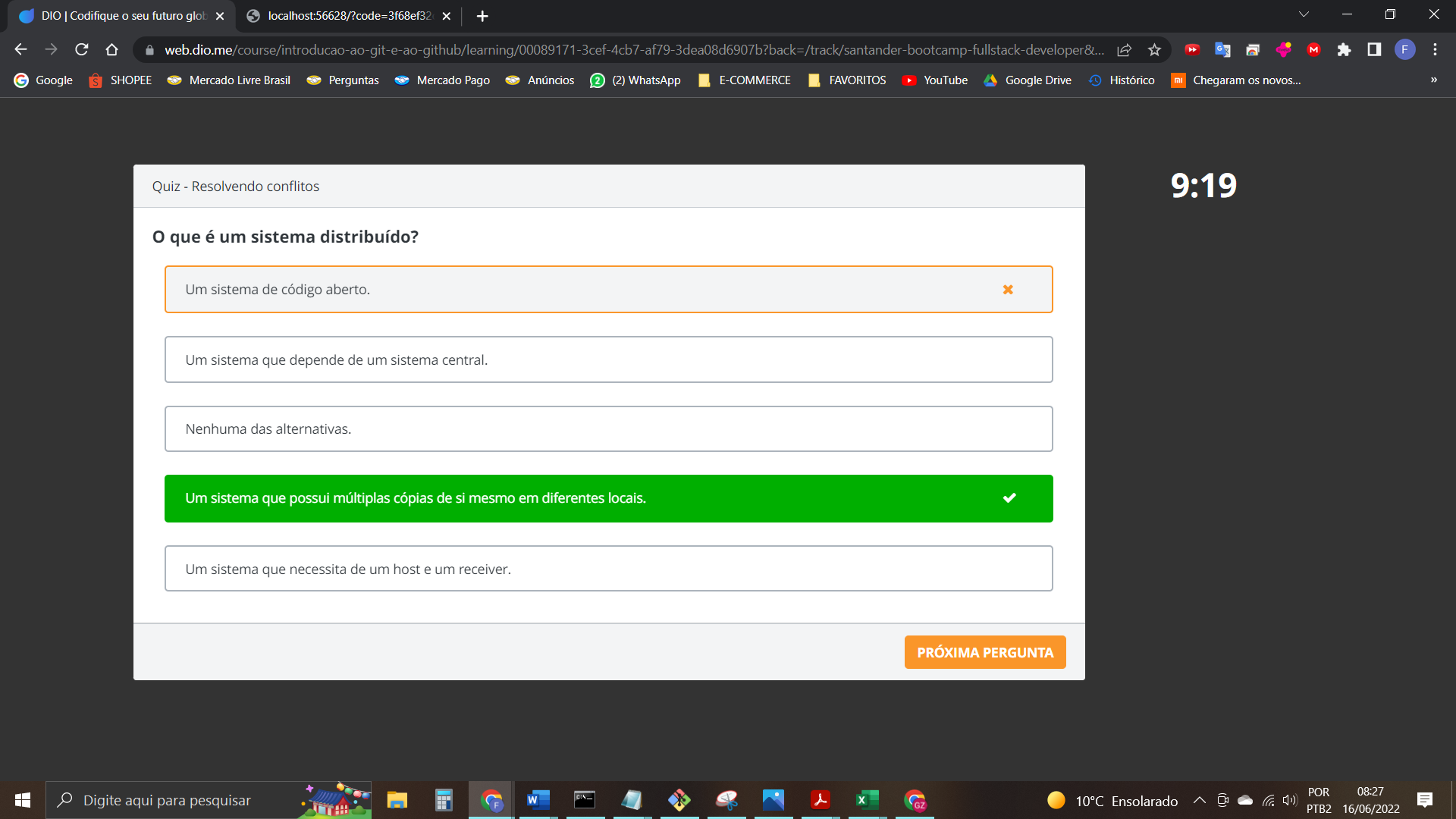




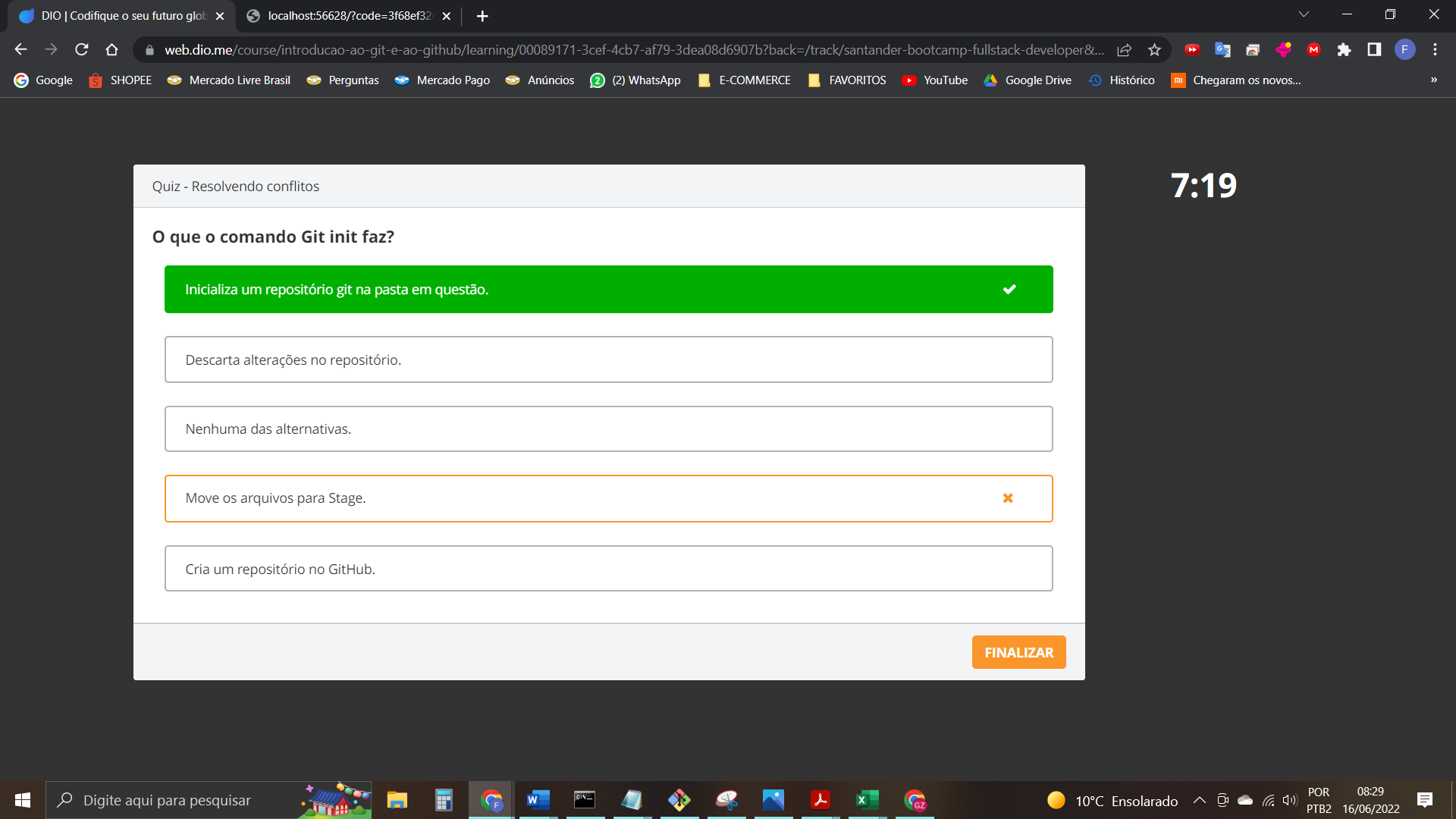












**Processo de apontamento do projeto após cria-lo no site do GitHub:**

**git remote add origin** <https://github.com/flaviotrz/htmlcss.git>

(faz o apontamento do caminho do projeto no github)

**git remote -v**

**git branch -M main**

**git push -u origin main**

**---------------------------------------------------------------**

**Ciclo de trabalho usual após fazer alterações nos arquivos:**

**git add .** adiciona pastas a stage para serem commitadas

**git commit -m** "Inclusão das Anotações do curso de Git/GitHub"

**git push origin main** empurra as alterações para o GitHub

Curso de Git no e-programar

<https://www.youtube.com/watch?v=2y5JGW6nZRs&list=PLM8_o_MDe-LHLvbzEJc89_4v3RA26p5AG>