

Introdução e Conceitos

1. Por que o Git é considerado um sistema de controle de versão distribuído?

r: porque cada desenvolvedor possui uma cópia local completa de todo o histórico do repositório, permitindo trabalhar e fazer commits offline sem depender de um servidor central.

2. Qual a diferença entre working directory, staging area e repository?

r: o working directory é a pasta com os arquivos atuais que você está editando, a staging area é uma área intermediária onde você agrupa as mudanças (git add) que irão para o próximo commit e o repository é onde o Git armazena permanentemente todo o histórico de commits (dentro da pasta .git).

3. Para que serve o comando git clone ?

r: para criar uma cópia de um repositório Git remoto (ex: do GitHub).

4. Onde estão implementados fisicamente working directory, staging area e repository ?

r: o working directory é a pasta principal do meu projeto, já a staging area e o repository ambos ficam armazenados dentro da subpasta oculta .git

5. Quais os estados de um arquivo no repositório do git ?

r: untracked, modified, Staged e Unmodified

6. Explique as possíveis transições de estado de um arquivo no repositório do git ?

r: um arquivo novo (Untracked) é adicionado (git add) e se torna Staged. após git commit, ele vira Unmodified. Se for editado, passa a Modified. ao usar git add nele, volta a ser Staged.

Prática com Git Local

Etapa 1 – Criar o repositório

Qual foi a mensagem exibida após o comando git init e o que ela significa na prática?

r: Initialized empty Git repository in C:/Users/Gustavo Dutra/aula-git/.git/ , significa que o Git criou uma nova subpasta oculta chamada .git dentro da pasta aula-git

Etapa 2 – Adicionar arquivo e fazer commit

1. Qual o estado do arquivo antes e depois do git add ?

r: antes do git add o arquivo estava no estado untracked (não rastreado), pois tinha acabado de ser criado e o Git ainda não o monitorava. depois do git add o arquivo passou para o estado staged (preparado), indicando que ele foi adicionado à staging area e está pronto para ser

incluído no próximo commit.

2. O que significa o estado untracked e tracked ?

r: untracked (Não rastreado) é um arquivo novo que está na pasta do projeto, mas que o Git ainda não monitora. o Git ignora mudanças feitas nele até você usar git add. o tracked (Rastreado) é um arquivo que o Git já "conhece" e monitora. Ele pode estar em três sub-estados: modified (modificado), staged (preparado) ou unmodified (não modificado).

3. Qual o objetivo do git commit ?

r: o objetivo é salvar permanentemente as mudanças que estão na staging area (os arquivos adicionados com git add) dentro do histórico do repositório local (a pasta .git).

4. Qual o estado do arquivo após o git commit ?

r: após o commit, o arquivo passa do estado staged para unmodified (não modificado). Isso significa que ele agora está salvo no repositório e a versão no working directory está idêntica à versão do último commit.

Etapa 3 – Histórico e alterações

1. O que o comando git diff mostra?

r: o git diff mostra as diferenças entre os arquivos no seu working directory e o que está no último commit. neste caso, ele vai mostrar que a "Nova linha" foi adicionada ao arquivo.txt, pois essa mudança ainda não foi salva em um commit.

2. O que acontece se você rodar git merge nova-feature estando na branch principal?

r: o HEAD está apontando para o "Primeiro commit", que foi o último commit realizado na Etapa 2. o HEAD é o ponteiro que indica onde você está trabalhando atualmente no histórico.

Etapa 4 – Trabalhando com Branches

1. Como verificar em qual branch você está?

r: voce pode usar o git branch ou o git status.

2. O que acontece se você rodar git merge nova-feature estando na branch principal?

r: o arquivo.txt da branch principal será atualizado para incluir a "Linha da nova branch".

Conectando ao GitHub

1. O que significa o -u no comando git push -u origin main ?

r: o -serve para criar um "link" permanente entre a sua branch local (main) e a branch remota no origin.

2. Como verificar os remotes configurados no repositório?

r: usando o comando git remote -v. ele lista todos os "apelidos" dos repositórios remotos (como origin) e mostra as URLs para onde eles apontam.

Encerramento e Discussão

Qual etapa foi mais difícil?

r: 2 etapa foi a mais complicada.

Como o Git ajuda na colaboração?

r: em tudo, desde o cmd ate salvar repositorio offline

Que diferença faz ter um repositório remoto?

r: segurança, consegue trabalhar com outras pessoas, acessibilidade