

Aluno: Lucas Terra Cunha

RA: 322123117

Introdução e conceitos

1. Por que o Git é considerado um sistema de controle de versão distribuído?
Porque cada desenvolvedor possui uma cópia completa do repositório (histórico completo) localmente, não dependendo de um servidor central para operações básicas.

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

- **Working Directory:** arquivos atuais que você edita
- **Staging Area:** área intermediária onde prepara commits (index)
- **Repository:** banco de dados com todo histórico de versões (.git)

3. Para que serve o comando git clone ?

Cria uma cópia local completa de um repositório remoto, incluindo todo histórico e branches.

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

Working Directory: diretório do projeto (arquivos visíveis)

Staging Area: arquivo .git/index

Repository: pasta .git/ (objects, refs, HEAD, etc)

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

- **Untracked:** não rastreado pelo Git
- **Unmodified:** rastreado, sem alterações
- **Modified:** rastreado e modificado
- **Staged:** preparado para commit

6. Transições de estado?

- Untracked → Staged: git add
- Unmodified → Modified: editar arquivo
- Modified → Staged: git add
- Staged → Unmodified: git commit

- Unmodified → Untracked: git rm
- Staged → Modified: editar arquivo staged

Prática com GIT local

Etapa 1:

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

A mensagem que apareceu foi:

Initialized empty Git repository in /home/caetano/projetos/aula-git/.git/

Significa que o Git criou a pasta oculta .git/ no diretório, transformando-o em um repositório Git. Essa pasta armazena todo o histórico de versões e configurações. O repositório está vazio e pronto para receber commits.

Etapa 2:

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

Antes: Untracked (arquivo novo, Git não conhece) Depois: Staged (preparado para commit)

2. O que significa o estado untracked e tracked ?

Untracked: arquivo que o Git não está monitorando Tracked: arquivo que o Git já conhece e monitora as alterações

3. Qual o objetivo do git commit ?

Salvar permanentemente as mudanças da staging area no repositório, criando um ponto no histórico de versões com uma mensagem descriptiva.

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

Unmodified (ou committed) - o arquivo está salvo no repositório sem alterações pendentes.

Etapa 3:

1. O que o comando git diff mostra?

Mostra as diferenças entre o arquivo no working directory (com as alterações) e o último commit. No caso, exibe a linha "+ Nova linha" que foi adicionada ao arquivo.txt mas ainda não foi staged.

2. Qual commit está atualmente apontado por HEAD?

O HEAD aponta para o commit mais recente:

f7a3c81 Primeiro commit

Etapa 4:

1. Como verificar em qual branch você está?

Usa git branch (mostra todas as branches e marca a atual com *) ou git status (mostra a branch atual no início da mensagem).

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

O Git vai mesclar as alterações da nova-feature na branch principal. Como a nova-feature está apenas à frente (um commit a mais), será um fast-forward merge - a branch principal simplesmente avança para incluir o commit "Alteração na nova branch". O arquivo.txt na principal vai ter as duas linhas.

Conectando ao GitHub

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

O -u (ou --set-upstream) configura a branch local para rastrear a branch remota. Depois disso, você pode usar apenas git push ou git pull sem precisar especificar origin main toda vez.

2. Como verificar os remotes configurados no repositório?

Usa git remote -v para ver os remotes com suas URLs, ou apenas git remote para listar os nomes. No caso, vai mostrar origin apontando para o GitHub.