■ Prática Git e GitHub - Respondido

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

   Porque cada pessoa que clona o repositório tem uma cópia completa de todo o histórico do projeto, podendo trabalhar offline e depois sincronizar com os outros.

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

   O working directory é onde ficam os arquivos no computador, a staging area é onde colocamos os arquivos que queremos preparar para o commit, e o repository é onde o Git guarda o histórico e os commits.

3. Para que serve o comando git clone?

   Serve para copiar um repositório já existente (geralmente do GitHub) para o computador local.

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

   O working directory são os arquivos na pasta do projeto, a staging area fica dentro da pasta .git (em um índice), e o repository também está dentro da pasta .git.

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

   Untracked (não rastreado), Tracked (rastreado), Modified (modificado), Staged (preparado) e Committed (confirmado).

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

   Um arquivo novo começa como untracked, quando usamos git add ele vira staged, depois do git commit ele fica committed. Se editarmos o arquivo de novo, ele vira modified.

Etapa 1 – git init: Qual foi a mensagem exibida e o que ela significa?

Aparece algo como 'Initialized empty Git repository'. Isso quer dizer que o Git criou uma pasta .git e começou a rastrear aquele diretório como um repositório.

Etapa 2 – Estado do arquivo antes e depois do git add?

Antes do git add ele está untracked, e depois passa para staged.

O que significa untracked e tracked?

Untracked é quando o Git ainda não está rastreando o arquivo, tracked é quando ele já faz parte do controle de versão.

Qual o objetivo do git commit?

Salvar uma versão do projeto no histórico, criando um ponto de recuperação.

Qual o estado do arquivo após o git commit?

Ele fica committed, ou seja, salvo no histórico e sem modificações pendentes.

Etapa 3 – O que o comando git diff mostra?

Mostra as diferenças entre o conteúdo atual dos arquivos e o que está salvo no último commit ou na staging area.

Qual commit está atualmente apontado por HEAD?

O último commit feito na branch atual.

Etapa 4 – Como verificar em qual branch você está?

Com o comando git branch ou git status.

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

Ele une as alterações da branch nova-feature na branch principal (main).

Conectando ao GitHub – O que significa o -u no git push -u origin main?

O -u define a branch remota como padrão, então nos próximos push e pull não é preciso escrever tudo de novo.

Como verificar os remotes configurados no repositório?

Com o comando git remote -v.

Encerramento – Qual etapa foi mais difícil?

Criar e entender as branches, porque é fácil se confundir com as mudanças em cada uma.

Como o Git ajuda na colaboração?

Permite que várias pessoas trabalhem no mesmo projeto sem sobrescrever o código dos outros e mantendo histórico de mudanças.

Que diferença faz ter um repositório remoto?

Permite armazenar o projeto online, compartilhar com outros e manter backup.