**ATIVIDADES GIT**

**1. O que é o Git e qual é a sua principal função no controle de versão de código-fonte?**

O Git é um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005 por Linus Torvalds, o famoso criador do kernel do sistema operacional Linux. Um número impressionante de projetos de software depende do Git para controle de versão, incluindo projetos comerciais e de código-fonte aberto. gerenciar o código-fonte e seu histórico de mudanças usando ferramentas de linha de comandos de Git

fonte

<https://www.atlassian.com/br/git/tutorials/what-is-git#:~:text=O%20Git%20%C3%A9%20um%20projeto,e%20de%20c%C3%B3digo%2Dfonte%20aberto>.

**2. Como podemos criar um novo repositório Git em um projeto existente no seu sistema local?**

Para criar um novo repositório, você vai usar o comando git init. git init é um comando único que você usa durante a configuração inicial de um novo repositório. A execução desse comando cria um novo subdiretório .git no diretório de trabalho atual. Essa ação também vai criar uma ramificação principal.

fonte

<https://www.atlassian.com/br/git/tutorials/setting-up-a-repository#:~:text=Para%20criar%20um%20novo%20reposit%C3%B3rio,comando%20cria%20um%20novo%20subdiret%C3%B3rio%20>.

**3. Qual a diferença entre "git add" e "git commit" no contexto do Git?**

**git add** significa **Adicionar**, ou seja, você pode adicionar um arquivo qualquer. Exemplo: git add nomearquivo.

Já em **git commit** é muito importante, pois indica as alterações que você fez no seu projeto. Ou seja, o commit é a **realização de um conjunto de mudanças, alterações que você realizou em seu projeto**.

fonte

<https://pt.stackoverflow.com/questions/572430/comandos-git-git-add-e-git-commit#:~:text=git%20add%20significa%20Adicionar%2C%20ou,voc%C3%AA%20realizou%20em%20seu%20projeto>.

**4. Explique o propósito do comando "git push" e quando você o utilizaria.**

O comando git push permite que você envie (ou em tradução literal, empurre) os commits de sua branch e repositório Git local para o seu repositório remoto. Para poder fazer um git push para seu repositório remoto, você deve garantir que todas as suas alterações no repositório local sejam confirmadas.

**fonte**

[**https://www.freecodecamp.org/portuguese/news/o-comando-git-push-explicado/#:~:text=O%20comando%20git%20push%20permite,no%20reposit%C3%B3rio%20local%20sejam%20confirmadas**](https://www.freecodecamp.org/portuguese/news/o-comando-git-push-explicado/#:~:text=O%20comando%20git%20push%20permite,no%20reposit%C3%B3rio%20local%20sejam%20confirmadas)**.**

**5. O que são branches no Git e por que são úteis em um projeto de desenvolvimento?**

Um branch no Git é simplesmente um ponteiro móvel para um desses commits. O nome do branch padrão no Git é master . Conforme você começa a fazer commits, você recebe um branch master que aponta para o último commit que você fez. Cada vez que você faz um novo commit, ele avança automaticamente.

fonte

<https://git-scm.com/book/pt-br/v2/Branches-no-Git-Branches-em-poucas-palavras#:~:text=Um%20branch%20no%20Git%20%C3%A9,novo%20commit%2C%20ele%20avan%C3%A7a%20automaticamente>.

**6. Como podemos verificar o status atual do repositório local no Git?**

O comando git status exibe as condições do diretório de trabalho e da área de staging. Ele permite que você veja quais alterações foram despreparadas, quais não foram e quais arquivos não estão sendo monitorados pelo Git.

fonte

<https://www.atlassian.com/br/git/tutorials/inspecting-a-repository#:~:text=O%20comando%20git%20status%20exibe,est%C3%A3o%20sendo%20monitorados%20pelo%20Git>.

**7. Explique o que é um conflito de merge no Git e como você resolveria esse problema.**

Os conflitos de merge ocorrem quando alterações concorrentes são feitas na mesma linha de um arquivo ou quando uma pessoa edita um arquivo e outra pessoa exclui o mesmo arquivo.

Você só pode resolver conflitos de merge no GitHub causados por alterações concorrentes na linha, como quando as pessoas fazem alterações diferentes na mesma linha do mesmo arquivo em diferentes branches no seu repositório Git. Para todos os outros tipos de conflito de merge, você deve resolver o conflito localmente na linha de comando.

fonte

<https://docs.github.com/pt/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-on-github>

**8. O que é o GitHub e como ele se diferencia do Git em termos de funcionalidades?**

O Github é uma “rede social dev” em que é possível armazenar e compartilhar projetos de desenvolvimento de software. O Git é um sistema de controle de versão de arquivos; em outras palavras, é responsável por guardar o histórico de alterações sempre que alguém modificar algum arquivo que está sendo monitorado por ele.

**9. Qual a diferença entre um repositório público e privado no GitHub?**

Com o GitHub Free para contas pessoais e de organizações, você pode trabalhar com colaboradores ilimitados em repositórios públicos ilimitados, com um conjunto completo de recursos, ou em repositórios privados ilimitados, com um conjunto limitado de recursos.

fonte

<https://docs.github.com/pt/repositories/creating-and-managing-repositories/about-repositories#:~:text=Com%20o%20GitHub%20Free%20para,um%20conjunto%20limitado%20de%20recursos>.

**10. Como podemos realizar o processo de clonagem (clone) de um repositório do GitHub para o seu computador local?**

Abra TerminalTerminalGit Bash. Crie um clone usando o comando git clone e o nome do repositorio.

fonte

<https://docs.github.com/pt/repositories/creating-and-managing-repositories/duplicating-a-repository#:~:text=Large%20File%20Storage-,Abra%20TerminalTerminalGit%20Bash.,reposit%C3%B3rio%20que%20voc%C3%AA%20deseja%20duplicar>.

**11.O que são Pull Requests no GitHub e qual é o seu propósito no desenvolvimento colaborativo?**

Os modelos de pull request permitem personalizar e padronizar as informações que você gostaria de incluir quando alguém cria uma pull request em seu repositório.

fonte

<https://docs.github.com/pt/pull-requests/collaborating-with-pull-requests/getting-started/best-practices-for-pull-requests#:~:text=Os%20modelos%20de%20pull%20request,pull%20request%20em%20seu%20reposit%C3%B3rio>.

**12. Como podemos adicionar um arquivo ao arquivo .gitignore e qual é a sua finalidade?**

Para sempre ignorar um determinado arquivo ou diretório, adicione-o a um arquivo chamado ignore que está localizado dentro do diretório ~/.config/git.

fonte

<https://docs.github.com/pt/get-started/getting-started-with-git/ignoring-files#:~:text=Para%20sempre%20ignorar%20um%20determinado,config%2Fgit%2Fignore%20>.

**13. Explique o que é um fork no GitHub e em que situações você pode querer realizar um fork de um repositório**.

Os forks permitem fazer alterações em um projeto sem afetar o repositório original, também conhecido como repositório "upstream".

Os forks geralmente são usados para iterar ideias ou alterações antes de serem propostas de volta para o repositório upstream, como em projetos código aberto ou quando um usuário não tem acesso de gravação ao repositório upstream. Para saber mais, confira "Trabalhar com bifurcações".

fonte

<https://docs.github.com/pt/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo#:~:text=Os%20forks%20geralmente%20s%C3%A3o%20usados,confira%20%22Trabalhar%20com%20bifurca%C3%A7%C3%B5es%22>.

**14. Como podemos visualizar o histórico de commits em um repositório Git?**

A ferramenta mais básica e poderosa para fazer isso é o comando git log . Por padrão, sem argumentos, git log lista os commits feitos neste repositório em ordem cronológica inversa; isto é, o commit mais recente aparece primeiro

fonte

<https://git-scm.com/book/pt-br/v2/Fundamentos-de-Git-Vendo-o-hist%C3%B3rico-de-Commits#:~:text=A%20ferramenta%20mais%20b%C3%A1sica%20e,%C3%A9%20o%20comando%20git%20log%20.&text=Por%20padr%C3%A3o%2C%20sem%20argumentos%2C%20git,commit%20mais%20recente%20aparece%20primeiro>.

**15.Qual é a função do comando "git pull" e quando ele deve ser utilizado no desenvolvimento colaborativo com o Git e GitHub?**

O comando git pull serve para recuperar e baixar o conteúdo de um repositório remoto e atualizar o repositório local assim que ele for baixado. Isso é muito útil em fluxos de trabalho de colaboração que precisam mesclar alterações upstream remotas no repositório local.

fonte

<https://www.locaweb.com.br/blog/temas/codigo-aberto/git-pull-aprenda-a-usar-o-comando/#:~:text=O%20comando%20git%20pull%20serve,upstream%20remotas%20no%20reposit%C3%B3rio%20local>.