Atividade dia 22/01/2024

Respostas:

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

R:

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.

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.

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

R:

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.

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

R:

Git add significa adicionar, ou seja, você pode adicionar um arquivo qualquer. Exemplo: git add “nomedomeuarquivoaqui”.

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.

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

R:

O Git push é um comando que possibilita que as alterações da sua máquina local sejam enviadas para uma máquina remota.

Fonte:

https://blog.betrybe.com/git/git-push/#1

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

R:

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.

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

R:

Inicialmente se estamos trabalhando em um arquivo local e temos um repositorio github teriamos que saber se a versão local e do repositório estão alinhadas. Para isso usamos: git status.

Fonte:

https://cursos.alura.com.br/forum/topico-comando-git-297811#:~:text=Inicialmente%20se%20estamos%20trabalhando%20em,git%20status

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

R:

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.

Para resolver um conflito de merge causado por alterações diferentes na linha, você deve escolher quais alterações dos diferentes branches incorporar em um novo commit.

Fonte:

https://docs.github.com/pt/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-using-the-command-line

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

R:

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.

Fonte:

https://www.alura.com.br/artigos/o-que-e-git-github#:~:text=O%20Github%20%C3%A9%20uma%20%E2%80%9Crede,est%C3%A1%20sendo%20monitorado%20por%20ele.

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

R:

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

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

R:

Para clonar um repositório localmente, use o subcomando repo clone . Substitua o parâmetro repository pelo nome do repositório.

Fonte:

https://docs.github.com/pt/repositories/creating-and-managing-repositories/cloning-a-repository

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

colaborativo?

R:

O pull request coloca para avaliação um fork, uma ramificação do projeto principal que permite alterações sem modificar o código principal, que passa por uma revisão antes de ser incorporado ao arquivo principal. Esse comando é realizado por meio da plataforma GitHub, e não direto no terminal.

Fonte:

https://www.locaweb.com.br/blog/temas/codigo-aberto/git-pull-aprenda-a-usar-o-comando/#:~:text=O%20pull%20request%20coloca%20para,e%20n%C3%A3o%20direto%20no%20terminal.

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

R:

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

Por padrão, o Git ignorará todos os arquivos e diretórios listados no arquivo de configuração global ~/. config/git/ignore .

Fonte:

https://docs.github.com/pt/get-started/getting-started-with-git/ignoring-files

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

um repositório.

R:

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.

Fonte:

https://docs.github.com/pt/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo

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

R:

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?

R:

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.