

Pesquisa sobre Git e Github

1- Git é um sistema de controle de versões distribuído, e pode ser usado para manter um histórico de alterações dos projetos em desenvolvimento. Sua principal função é registrar todas as alterações feitas no código-fonte.

2- Para criar um novo repositório vá até a pasta em que está seu projeto e execute o comando "git init".

3- A diferença é que o "git add" é usado para adicionar um arquivo ao "staging area", onde você prepara esse arquivo para ser "commitado". Usando o "git commit" você cria um commit com uma mensagem descritiva do que foi alterado no projeto.

4- O comando "git push" é utilizado para enviar seu projeto "commitado" para o repositório remoto. É utilizado para fazer atualização do projeto no repositório remoto.

5- Branches são ramificações do código-fonte que cria uma linha independente de desenvolvimento. O recurso funciona como um controle de origem que permite separar a evolução da programação em diferentes subdivisões. São úteis para testar novas funcionalidades sem afetar o código principal, para fazer a correção de bugs encontrados no código principal.

6- Usando o comando "git status" podemos verificar o status do repositório local.

7- 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. Para resolver um conflito de merge causado por alterações concorrentes em um arquivo, quando uma pessoa exclui um arquivo em um branch e outra pessoa edita o mesmo arquivo, você deve escolher se deseja excluir ou manter o arquivo removido em um novo commit. Por exemplo, se você editar um arquivo, como LEIAME.md, e outra pessoa remover o mesmo arquivo em outro branch no mesmo repositório Git, você receberá um erro de conflito de mesclagem ao tentar mesclar esses

branches. Você deve resolver esse conflito de merge com um novo commit antes de fazer merge desses branches.

8- GitHub é uma plataforma para gerenciar seu código e criar um ambiente de colaboração entre devs, utilizando o Git como sistema de controle. O git é um software VCS local que permite aos desenvolvedores salvarem snapshots de seus projetos ao longo do tempo. Geralmente é melhor para uso individual. GitHub é uma plataforma baseada na web que incorpora os recursos de controle de versões do git para que possam ser usados colaborativamente.

9- A diferença é que em repositório público qualquer um pode ver, forkar e baixar o código fonte. Enquanto em projetos privados apenas os usuários que você liberar terão acesso ao código.

10- Primeiro você irá copiar a url do repositório que deseja clonar, depois abra o Git Bash e altere o diretório de trabalho atual para o local, após isso digite "git clone" e cole a url copiada.

11- Uma pull request é uma proposta para mesclar as alterações de um branch em outro. Seu principal propósito é permitir que os colaboradores revisem e discutam o conjunto de alterações proposto antes de integrá-las à base de código principal.

12- Para criar um arquivo .gitignore local, crie um arquivo de texto e dê a ele o nome de .gitignore (lembre-se de incluir o "." no começo). Em seguida, edite esse arquivo conforme necessário. Cada nova linha deve listar um arquivo ou pasta adicional que você quer que o Git ignore.

As entradas neste arquivo também podem seguir um padrão de correspondência.

* -> é usado para corresponder a um caractere curinga

/ -> é usado para ignorar nomes de caminhos relativos ao arquivo .gitignore

-> é usado para adicionar comentários ao arquivo .gitignore

A sua finalidade é dizer ao Git quais arquivos ou pastas ele deve ignorar em um projeto. Também é possível criar um arquivo global com o comando "git config -global core.excludesfile ~/.gitignore_global".

Isso criará o arquivo ~/.gitignore_global. Agora, você pode editar esse arquivo do mesmo modo que faz com um arquivo .gitignore local. Todos os seus repositórios do Git vão ignorar os arquivos e pastas listados no arquivo .gitignore global.

13- Os forks permitem fazer alterações em um projeto sem afetar o repositório original, também conhecido como repositório "upstream". Após criar fork em um repositório, você pode buscar atualizações do repositório upstream para manter seu fork atualizado e propor alterações do fork para o repositório upstream com solicitações de pull. Um fork pode ser propriedade de uma conta pessoal ou de uma organização. A sua finalidade é permitir que ideias ou alterações sejam testadas antes de serem incorporadas ao repositório upstream.

14- Usando o comando "git log", esse comando lista todos os commits feitos em ordem cronológica inversa, ou seja, o commit mais recente é exibido em primeiro lugar.

15- O comando git pull é usado para buscar e baixar conteúdo de repositórios remotos e fazer a atualização imediata ao repositório local para que os conteúdos sejam iguais. Ele deve ser usado quando uma nova atualização é feita no repositório upstream, para que seu projeto no repositório local fique atualizado.