

## Desafio: hora da prática.

O Git é uma ferramenta excelente para acompanhar a mudança entre versões de um mesmo projeto e, dentre vários benefícios, nos ajuda a observar de perto o desenvolvimento do seu aprendizado. Tudo isso de uma forma organizada através dos commits.

Além disso, algo que é essencial no universo da tecnologia é apresentar o seu progresso para a comunidade e montar um portfólio dos seus projetos para demonstrar suas habilidades. Dessa forma, o GitHub é essencial para compartilhar e colaborar em projetos de programação de todo o mundo.

Pensando nisso, criamos uma lista de atividades (não obrigatórias) focada em prática para melhorar ainda mais sua experiência de aprendizagem. Bora praticar então?

### Desafios

1. Crie uma conta no GitHub
  2. Crie um repositório para um projeto pessoal.
  3. Faça a instalação do Git
  4. Crie um repositório localmente para o seu projeto pessoal
  5. Adicione alguns arquivos no seu repositório local
  6. Faça o commit das alterações
  7. Configure a identidade do autor do commit.
  8. Crie a branch `Main`
  9. Realize a conexão do seu repositório local com o remoto
  10. Envie as alterações no repositório local para o remoto
  11. Utilize o comando `git status`
-

Caso precise de ajuda, opções de solução das atividades estarão disponíveis na seção “Opinião da pessoa instrutora”.

### Opinião do instrutor

1. Crie uma conta no GitHub Você pode começar seguindo os passos mostrados no vídeo, preenchendo o formulário de cadastro, verificando sua conta e explorando a página inicial do GitHub.
2. Criar um novo repositório no GitHub e fazer o upload de um projeto local para esse repositório. Você pode seguir passos mencionados na aula, como acessar as configurações do GitHub, criar um novo repositório com um nome único e escolher se ele será público ou privado. Em seguida, você pode adicionar uma descrição, um README, um `.gitignore` e uma licença ao repositório.
3. Instalação do Git: Caso você ainda não tenha realizado a instalação, siga os passos na atividade [Faça como eu fiz: instalação do Git](#)
4. Para criar um repositório local você, digite o seguinte comando no terminal: `git init`
5. Para adicionar os arquivos no repositório local, digite no terminal o comando: `git add .`
6. Faça um commit com as modificações, digite no terminal o comando: `git commit -m "mensagem de commit"`
7. Para configurar a identidade do autor do commit, digite no terminal o comando:
8. `git -config --global user.email "seuemailaquiaqui@example.com"`
9. `git config --global user.name "seu nome aqui"`
10. Para criar a branch Main, digite no terminal o comando: `git branch -M main`

O comando `git branch -m` é usado para criar uma nova ramificação no repositório Git atual. Neste caso, criamos a branch padrão `main`, que representa a versão principal do código.

11. Para realizar a conexão do seu repositório local com o remoto via SSH, digite no terminal: `git remote add origin`

`https://github.com/seunomedeusuario/seu-repositorio.git`

Caso seja necessário, realize a configuração do protocolo SSH através da geração de chave, você pode acompanhar os passos em vídeo na atividade [Sincronizando repositórios](#)

12. Para subir as alterações no repositório local para o remoto, digite o seguinte comando no terminal: `git push -u origin main`

13. Digite no terminal o comando `git status` e observe a saída.

O comando `git status` é uma ferramenta essencial para gerenciar alterações no controle de versão Git. Ele fornece uma visão geral do estado atual do repositório, indicando quais arquivos foram modificados, adicionados ou excluídos desde o último commit. Essa informação é crucial para compreender o progresso do desenvolvimento e tomar decisões de gerenciamento de alterações.

A saída do comando `git status` geralmente contém três seções principais:

**Modificados:** Lista os arquivos que foram modificados desde o último commit, mas ainda não foram adicionados à área de preparação (Stage).

**Adicionados:** Indica os arquivos que foram adicionados à área de preparação, mas ainda não foram confirmados no histórico de commits.

**Modificados, adicionados ou excluídos:** Exibe os arquivos que não foram rastreados pelo Git, ou seja, que não foram adicionados ao índice de modificações (Staging Area).

Além disso, o comando `git status` pode fornecer informações adicionais sobre as ramificações atuais, como a ramificação atual e as ramificações que estão à frente ou atrás da atual.

O comando `git status` é uma ferramenta indispensável para qualquer desenvolvedor que utiliza Git. Ele permite monitorar as alterações no repositório, identificar o estado dos arquivos e tomar decisões de gerenciamento de commits de forma eficiente.