

## Desafios: hora da prática

Conflitos em diferentes versões no código é algo comum no cotidiano da pessoa desenvolvedora e é essencial saber como resolver quando esses conflitos aparecem. A ideia do próximo desafio é gerar e resolver um conflito entre um repositório remoto e local, vamos praticar com os desafios não obrigatórios?

### Desafios

- Crie um novo repositório local
  - Adicione o repositório remoto criado nos exercícios anteriores ao seu repositório local.
  - Faça uma alteração no repositório local e envie para o remoto.
  - Resolva os conflitos manualmente, escolhendo quais alterações serão mantidas
  - Realize um commit para registrar a resolução do conflito.
  - Verifique quais arquivos foram adicionados
  - Sincronize o repositório local com o repositório remoto no GitHub.
- 

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

### Opinião do instrutor

#### Desafio 1: Criar um Novo Repositório Local

- Abra o terminal.
- Navegue até o diretório onde deseja criar o novo repositório.
- Execute `git init` para inicializar um novo repositório.

#### Desafio 2: Adicionar o Repositório Remoto Criado Anteriormente

- Execute `git remote add <nome-remoto> <url-do-repositorio-remoto>` para adicionar o repositório remoto ao seu local.

### **Desafio 3: Faça uma Alteração no Repositório Local e Envie para o Remoto**

- Abra um arquivo existente ou crie um novo no seu repositório local.
- Faça as alterações desejadas.
- Execute `git add .` para adicionar as alterações.
- Em seguida, execute `git commit -m "Alteração no repositório local"` para realizar o commit.
- Utilize `git push <nome-remoto> main` para enviar as alterações para o repositório remoto.

### **Desafio 4: Resolvendo Conflitos Manualmente**

- No GitHub, faça uma alteração no mesmo arquivo que foi modificado localmente.
- Execute `git pull <nome-remoto> main` no seu terminal para trazer as alterações remotas.
- O Git indicará um conflito. Abra o arquivo afetado e resolva manualmente, escolhendo quais alterações manterá.

Sugestão: você pode seguir os passos demonstrados no vídeo [Resolvendo conflitos](#)

### **Desafio 5: Realize um Commit para Registrar a Resolução do Conflito**

- Após resolver manualmente, execute `git add .` para adicionar as alterações ao *staged area*.
- Em seguida, execute `git commit -m "Resolver conflito manualmente de 'descrição do conflito'"` para registrar a resolução do conflito.

### **Desafio 6: Verifique Quais Arquivos Foram Adicionados**

- Execute `git status` para verificar se há alterações pendentes.
- Se quiser ver as diferenças, utilize `git diff`.
- Conheça mais sobre o comando `git diff` na [documentação git diff](#)

**\*\*Desafio 7: Sincronize o Repositório Local com o Repositório Remoto no GitHub\*\***

- Execute `git push <nome-remoto> main` para enviar as alterações, incluindo a resolução de conflitos, para o repositório remoto no GitHub.
- Verifique o histórico de commits no GitHub e confirme que o conflito foi resolvido corretamente.

Parabéns! Você concluiu o desafio de trabalhar com conflitos no Git, resolvendo manualmente e sincronizando seu repositório local com o remoto. Este processo é crucial ao colaborar em projetos com outros desenvolvedores.