

ATIVIDADE: Enviar dados locais para o Remoto

"Usuário A" possui <https://github.com/BSI-UFU/TMS.git> e "Usuário B" (o colega) possui https://github.com/Teste_Manutencoes/TMS.git.

Os usuários já clonaram seus próprios repositórios (que se torna o origin padrão) e adicionaram o repositório do colega como um segundo remoto , usando:

`git remote add <nome> <url>.`

- Usuário A (BSI-UFU) executa:
Clona o próprio repositório
`git clone https://github.com/BSI-UFU/TMS.git`
`cd TMS`
Adiciona o repositório do Usuário B como um remote chamado "colegaB"
`git remote add colegaB https://github.com/Teste_Manutencoes/TMS.git`
- Usuário B (Teste_Manutencoes) executa:
Clona o próprio repositório
`git clone https://github.com/Teste_Manutencoes/TMS.git`
`cd TMS`
Adiciona o repositório do Usuário A como um remote chamado "colegaA"
`git remote add colegaA https://github.com/BSI-UFU/TMS.git`

Executando a Atividade

Aqui estão os passos da atividade, executados pelo Usuário A:

1. Verifique o estado do repositório

Antes de qualquer mudança, o Usuário A verifica o estado local.

(No terminal do Usuário A)

Verifica o status (deve estar limpo)

`$ git status`

On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Lista os remotos configurados com suas URLs

`$ git remote -v`

colegaB https://github.com/Teste_Manutencoes/TMS.git (fetch)

colegaB https://github.com/Teste_Manutencoes/TMS.git (push)

origin <https://github.com/BSI-UFU/TMS.git> (fetch)

origin <https://github.com/BSI-UFU/TMS.git> (push)

ATIVIDADE: Enviar dados locais para o Remoto

Busca atualizações de todos os remotos

```
$ git fetch --all
```

Lista os ramos locais e seus status de rastreamento

```
$ git branch -vv
```

```
* main 1ae2a45 [origin/main] initial commit
```

Observação: O estado inicial está limpo e sincronizado com o origin do Usuário A.

2. Faça "commit" localmente e envie ao repositório remoto

O Usuário A faz uma mudança, faz o "commit" e envia para seu próprio repositório remoto (origin).

(No terminal do Usuário A)

1. Cria ou edita um arquivo

```
$ echo "Mudanca inicial do Usuario A" > readme.md
```

2. Adiciona e faz o commit

```
$ git add readme.md
```

```
$ git commit -m "Adiciona readme pelo Usuario A"
```

3. Envia os dados locais para o ramo 'main' do remoto 'origin'

```
$ git push origin main
```

Observação: As mudanças do Usuário A estão agora em <https://github.com/BSI-UFU/TMS.git>.

3. Solicite ao seu colega (Usuário B) que atualize o repositório dele

O Usuário B (o colega) agora busca as alterações do Usuário A (do remote colegaA).

(No terminal do Usuário B)

1. Busca os dados do remote 'colegaA' (BSI-UFU) [cite: 130]

Isso recupera os dados, mas não mescla [cite: 143]

```
$ git fetch colegaA
```

2. Mescla as mudanças do ramo 'main' de 'colegaA' no 'main' local

```
$ git merge colegaA/main
```

Observação: O Usuário B agora tem as mudanças do Usuário A em seu repositório local.

4. Solicite que seu colega (Usuário B) realize mudanças e envie ao remoto

O Usuário B edita o arquivo, faz o "commit" e envia para seu remoto (origin para o Usuário B).

ATIVIDADE: Enviar dados locais para o Remoto

(No terminal do Usuário B)

1. Edita o arquivo

```
$ echo "Adicionando contribuicao do Usuario B" >> readme.md
```

2. Adiciona e faz o commit

```
$ git add readme.md
```

```
$ git commit -m "Modifica readme pelo Usuario B"
```

3. Envia os dados para o 'origin' do Usuário B (Teste_Manutencoes)

```
$ git push origin main
```

Observação: As mudanças do Usuário B estão agora em https://github.com/Teste_Manutencoes/TMS.git.

5. Atualize seu repositório local com as mudanças (Usuário A)

O Usuário A agora busca as mudanças que o Usuário B acabou de enviar para o repositório colegaB.

(No terminal do Usuário A)

1. Busca os dados do remote 'colegaB' [cite: 130]

```
$ git fetch colegaB
```

2. Mescla as mudanças recebidas [cite: 675, 680]

```
$ git merge colegaB/main
```

Observação: O repositório local do Usuário A agora contém ambas as contribuições.

6. Observe as diferenças entre os repositórios

O Usuário A verifica o estado final.

(No terminal do Usuário A)

Verifica o status dos ramos locais em relação aos remotos [cite: 773, 793]

```
$ git branch -vv
```

```
* main 5f8a2d1 [origin/main: behind 1, colegaB/main] Merge remote-tracking branch 'colegaB/main'
```

Observação: O 'main' local está atualizado com 'colegaB/main',

mas está 1 commit "atrás" (behind) do 'origin/main'.

Isso é um erro na observação, deveria ser "ahead" (à frente).

Vamos corrigir: o local 'main' do Usuário A agora inclui o merge do 'colegaB',

então ele está *à frente* (ahead) do 'origin/main' do Usuário A.

\$ git branch -vv

ATIVIDADE: Enviar dados locais para o Remoto

```
# * main 5f8a2d1 [origin/main: ahead 1, colegaB/main] Merge remote-tracking branch  
'colegaB/main'
```

Para sincronizar tudo, o Usuário A envia a mesclagem para seu 'origin'

```
$ git push origin main
```

Verificando novamente após o push final

```
$ git fetch --all
```

```
$ git branch -vv
```

```
* main 5f8a2d1 [origin/main, colegaB/main] Merge remote-tracking branch  
'colegaB/main'
```

Observação Final: O main local do Usuário A, o origin/main (BSI-UFU) e o colegaB/main (Teste_Manutencoes) estão todos sincronizados e contêm as mudanças de ambos os usuários.