

Fetch

ATUALIZANDO REPOSITÓRIO DE JOÃO ATRAVÉS DO COMANDO FETCH

Acessar o repositório do **João**

Verificar o histórico da branch atual

```
$ git hist
```

Nota: este repositório local está desatualizado com os vários commits realizados por Maria

Realizar o download do conteúdo remoto

```
$ git fetch
```

Verificar o histórico da branch atual

```
$ git hist
```

Nota: o histórico permanece desatualizado, mesmo após a execução do comando fetch. Isso porque o histórico é da branch local

Listar arquivos e diretórios

```
$ ls
```

Verificar o histórico da branch de rastreamento origin/master

```
$ git hist origin/master
```

Nota: o histórico da branch master possui todos os commits que existem no Bitbucket

Realizar merge com a branch de rastreamento origin/master

```
$ git merge origin/master
```



SEÇÃO 6: TRABALHO COLABORATIVO

Nota: foi realizado um merge com a estratégia fast-forward

Verificar o histórico da branch atual

```
$ git hist
```

Nota: todos os commits foram incorporados na branch atual

Listar arquivos e diretórios

```
$ ls
```

Pergunta natural: Então sempre que for necessário atualizar a área de trabalho deve ser feito um fetch e um merge?

A resposta é sim. Mas existe um comando que realiza esses dois passos chamado git pull



Pull

GERANDO UM COMMIT NO REPOSITÓRIO DE JOÃO EM UMA NOVA BRANCH ROLES

Acessar o repositório de **João** via Git Bash

Criar uma nova branch roles e alternar para a mesma

```
$ git switch -c roles
```

Gerar um novo commit nesta nova branch que adiciona um novo arquivo roles.js

```
$ echo "function checkRole(){...}" >> roles.js  
$ git add roles.js  
$ git commit -m "add check role"
```

Enviar esta branch para o repositório remoto

```
$ git push -u origin roles
```

Acessar o Bitbucket e visualizar o histórico da branch master e roles

ATUALIZANDO REPOSITÓRIO DE MARIA ATRAVÉS DO COMANDO FETCH

Acessar o repositório de **Maria** via Git Bash, considerando que o objetivo dela é colaborar com alterações na branch roles

Realizar o download do conteúdo remoto

```
$ git fetch
```



Verificar todas as branches

```
$ git branch -a
```

Nota: não existe branch local roles, somente de rastreamento

CRIANDO BRANCH LOCAL ROLES E GERANDO UM NOVO COMMIT NO REPOSITÓRIO DE MARIA

Alternar para a branch roles

```
$ git switch roles
```

Nota: por existir uma branch de rastreamento com o nome roles, é criado a branch local que também é configurada para rastrear origin/roles

Gerar um novo commit nesta nova branch

```
$ echo "function createRole(){...}" >> roles.js  
$ git add roles.js  
$ git commit -m "add function to create a new role"
```

Enviar ao repositório remoto

```
$ git push
```

Nota: como git switch, neste caso, já configurou o rastreamento, não foi necessário -u origin roles

ATUALIZANDO REPOSITÓRIO DE JOÃO ATRAVÉS DO COMANDO PULL

No repositório de **João** verificar o histórico

```
$ git hist
```



Atualizar o repositório utilizando "pull"

```
$ git pull
```

Verificar o histórico novamente

```
$ git hist
```

Nota: o commit "add function to create a new role" foi adicionado ao histórico e a branch local roles é uma referência para este commit



Bloqueio de push

GERANDO UM COMMIT ATRAVÉS DA COLABORADORA MARIA

Acessar o repositório de **Maria** via Git Bash

Criar um novo arquivo e gerar um commit

```
$ echo "function createPrivilege(){...}" >> privileges.js  
$ git add privileges.js  
$ git commit -m "add function to create a new privilege"
```

Enviar ao repositório remoto

```
$ git push
```

ENTENDENDO CENÁRIO DE BLOQUEIO DE PUSH

No repositório de **João** gerar um novo commit, sem atualizar o repositório antes

```
$ echo "function removeRole(){...}" >> roles.js  
$ git add roles.js  
$ git commit -m "add function to remove a role"
```

Comparar o histórico da branch roles no ambiente local com o Bitbucket

```
$ git hist
```

Acessar o Bitbucket/Commits e filtrar pela branch roles

Tentar enviar ao repositório remoto

```
$ git push
```

Nota: de acordo com a mensagem exibida, o repositório está desatualizado e precisa ser atualizado antes de um push



SINCRONIZANDO O REPOSITÓRIO

Atualizar o repositório

```
$ git pull
```

***Nota:** foi necessário um merge do tipo three-way e por isso foi solicitado uma mensagem do commit de merge*

Verificar o histórico

```
$ git hist
```

Enviar ao repositório remoto

```
$ git push
```

Retornar ao repositório de **Maria** e atualizar

```
$ git pull
```

Verificar o histórico

```
$ git hist
```



Conflito entre os colaboradores

GERANDO UM COMMIT ATRAVÉS DA COLABORADORA MARIA

Acessar o repositório de **Maria** via Git Bash

Verificar o conteúdo do arquivo roles.js

```
$ cat roles.js
```

Nota: neste momento o arquivo possui 3 linhas

Gerar um novo commit, **que adiciona uma quarta linha**

```
$ echo "function editRole(){...}" >> roles.js  
$ git add roles.js  
$ git commit -m "add function to edit a role"
```

Enviar ao repositório remoto

```
$ git push
```

CONFLITO ENTRE OS COLABORADORES

No repositório de **João** verificar o conteúdo do arquivo roles.js

```
$ cat roles.js
```

Nota: neste momento o arquivo possui 3 linhas, pois ainda não houve atualização

Gerar um novo commit, **sem atualizar o repositório antes**

```
$ echo "function getRoles(){...}" >> roles.js  
$ git add roles.js  
$ git commit -m "add function to get roles"
```

Nota: como João não atualizou o repositório antes do commit, ele partiu de uma versão desatualizada do arquivo roles.js



SEÇÃO 6: TRABALHO COLABORATIVO

Tentar enviar ao repositório remoto

```
$ git push
```

Nota: de acordo com a mensagem exibida, o repositório está desatualizado e precisa ser atualizado antes de um push

Atualizar o repositório

```
$ git pull
```

Nota: no merge gerado pelo comando `git pull` houve conflito, pois João e Maria atualizaram a mesma linha de arquivo

RESOLVENDO O CONFLITO

Resolva o conflito com o editor de preferência

Adicionar a resolução do conflito na área de preparo e fazer o commit

```
$ git add roles.js  
$ git commit
```

Verificar o histórico

```
$ git hist
```

Enviar ao repositório remoto

```
$ git push
```

Retornar ao repositório de **Maria** e atualizar

```
$ git pull
```



Verificar o histórico

```
$ git hist
```



Pull request

CRIANDO UM PULL REQUEST

No Bitbucket, acessar a aba branches

Selecionar a branch roles enviada ao repositório remoto

Clicar na opção create pull request

Na nova tela, verifique os seguintes itens:

- Source Branch: trata-se da branch de origem
- Destination Branch: trata-se da branch de destino
- Description: trata-se da descrição do pull request
- Reviewers: campo que permite a adição dos revisores
- Delete <branch> after the pull request is merged: determina que após o merge para a branch de destino, a branch de origem será removida

Criar a solicitação de revisão

Nota: em um processo comum, o solicitante não consegue aprovar seu próprio pull request, mas isso depende das configurações do repositório no Bitbucket. Neste caso, isso não foi configurado e é possível aprovar a própria solicitação

Acessar a aba commits e filtrar pela branch master

Verificar a existência dos commits realizados nesta seção

ATUALIZANDO REPOSITÓRIOS LOCAIS

Retornar ao repositório local de Maria e alternar para a branch master

```
$ git switch master
```

Atualizar o repositório local com o merge que ocorreu no repositório remoto

```
$ git pull
```

Retornar ao repositório local de João e alternar para a branch master

```
$ git switch master
```



Atualizar o repositório local com o merge que ocorreu no repositório remoto

```
$ git pull
```



Está gostando deste curso?

*Compartilhe sua experiência nas redes sociais com a tag
#rsantanatech para que eu possa interagir com a sua postagem.*

**Acompanhe nas redes sociais
e fique por dentro de todos os conteúdos.**

