

Descartando alterações da área de trabalho

ENTENDENDO OS COMANDOS RESTORE E CLEAN

Acessar o repositório de Maria via Git Bash

Criar uma nova branch e alternar para a mesma

```
$ git switch -c login
```

Nota: a opção `-c` cria a branch com o nome "login" e alterna imediatamente para a branch recém criada

Verificar o estado do repositório

```
$ git status
```

Verificar o conteúdo do arquivo login.js

```
$ cat login.js
```

Adicionar uma nova linha no arquivo existente login.js

```
$ echo "function checkPassword(){...}" >> login.js
```

Verificar o estado do repositório

```
$ git status
```

Siga a orientação do próprio Git para descartar as mudanças locais (sugerido no comando acima)

```
$ git restore login.js
```



SEÇÃO 4: DESCARTANDO ALTERAÇÕES

Verificar o estado do repositório

```
$ git status
```

Verificar o conteúdo do arquivo login.js

```
$ cat login.js
```

Adicionar um novo arquivo logout.js

```
$ touch logout.js
```

Verificar o estado do repositório

```
$ git status
```

Nota: o estado deste arquivo é *untracked*

Tentar descartar este novo arquivo com git clean padrão

```
$ git clean logout.js
```

Nota: por padrão Git só irá remover se for utilizado a opção para forçar a exclusão (-f)

Descartar este novo arquivo utilizando a opção -f no git clean

```
$ git clean logout.js -f
```

Nota: cuidado ao realizar esse comando! Lembre-se que não há histórico de alterações desse arquivo

Nota: neste comando, se é desejável remover todos os arquivos não rastreados basta usar `$git clean -f` (não é necessário trocar o nome do arquivo por '.' para remover todos)



Descartando alterações da área de preparo

ENTENDENDO O COMANDO RESTORE COM A OPÇÃO --STAGED

Acessar o repositório de Maria via Git Bash

Adicionar uma nova linha no arquivo existente login.js

```
$ echo "function checkPassword(){...}" >> login.js
```

Adicionar um novo arquivo logout.js

```
$ touch logout.js
```

Adicionar as duas alterações para a área de preparo

```
$ git add .
```

Verificar o estado do repositório

```
$ git status
```

Siga a orientação do próprio Git para descartar as mudanças da área de preparo (sugerido no comando acima)

```
$ git restore --staged .
```

Verificar o estado do repositório

```
$ git status
```



Revert de commit

ENTENDENDO O COMANDO REVERT

Acessar o repositório de Maria via Git Bash

Verificar o estado do repositório

```
$ git status
```

Descartar a inclusão do novo arquivo logout.js

```
$ git clean logout.js -f
```

Realizar o commit da alteração do arquivo login.js

```
$ git add login.js  
$ git commit -m "add check password"
```

Verificar o conteúdo do arquivo login.js

```
$ cat login.js
```

Verificar o histórico da branch atual

```
$ git hist
```

Desfaça o último commit

```
$ git revert head
```

Nota: será aberto o editor padrão para informar a mensagem de reversão. É interessante que o novo commit de uma reversão utilize a mensagem padrão "revert <mensagem do commit original>"



SEÇÃO 4: DESCARTANDO ALTERAÇÕES

Nota: a opção `--no-edit` poderia ser utilizada neste comando, que iria manter o padrão acima, sem abrir o editor de texto padrão

Nota: este comando pode receber um id de commit ou qualquer referência tag, branch ou HEAD

Nota: neste exemplo foi utilizado a referência HEAD, que indiretamente era uma referência para o último commit da branch atual. Esse último commit que foi revertido. Para reverter o penúltimo commit pode-se utilizar: `$ git revert HEAD~1`

Verificar o histórico da branch atual

```
$ git hist
```

Verificar o conteúdo do arquivo login.js

```
$ cat login.js
```



Reset – soft

ENTENDENDO O COMANDO RESET COM A OPÇÃO --SOFT

Acessar o repositório de Maria via Git Bash

Adicionar uma nova linha no arquivo existente login.js

```
$ echo "function validate(){...}" >> login.js
```

Gerar um commit

```
$ git add login.js  
$ git commit -m "add login validate"
```

Verificar o histórico da branch atual

```
$ git hist
```

Realizar um reset para o penúltimo commit utilizando a opção --soft

```
$ git reset --soft head~
```

Verificar o histórico da branch atual

```
$ git hist
```

Verificar o estado do repositório

```
$ git status
```

Nota: após um reset para um commit anterior (Revert "add check password"), o Git está solicitando o commit da alteração anterior. Isso ocorreu porque com a opção soft as áreas de preparo e área de trabalho não são alteradas



Reset – mixed

ENTENDENDO O COMANDO RESET COM A OPÇÃO --MIXED

Acessar o repositório de Maria via Git Bash

Gerar novamente o commit que foi eliminado do histórico na aula anterior

```
$ git commit -m "add login validate"
```

Verificar o histórico da branch atual

```
$ git hist
```

Realizar um reset para o penúltimo commit utilizando a opção --mixed

```
$ git reset head~
```

Nota: não é necessário incluir --mixed, pois o mixed é a opção padrão

Verificar o histórico da branch atual

```
$ git hist
```

Verificar o estado do repositório

```
$ git status
```

Nota: após um reset para um commit anterior (Revert "add check password"), o Git está informando que a alteração do arquivo login.js pode ser adicionada a área de preparo. Isso ocorreu porque com a opção mixed a área de preparo é alterada para o momento do commit selecionado, que não tinha a função validate



Reset – hard

ENTENDENDO O COMANDO RESET COM A OPÇÃO --HARD

Acessar o repositório de Maria via Git Bash

Gerar novamente o commit que foi eliminado do histórico na aula anterior

```
$ git add login.js  
$ git commit -m "add login validate"
```

Verificar o histórico da branch atual

```
$ git hist
```

Realizar um reset para o penúltimo commit utilizando a opção --hard

```
$ git reset --hard head~
```

Verificar o histórico da branch atual

```
$ git hist
```

Verificar o estado do repositório

```
$ git status
```

Nota: após um reset para um commit anterior (Revert “add check password”), o Git está informando que o diretório está limpo, ou seja, sem alterações para serem versionadas. Isso ocorreu porque com a opção hard a área de preparo e a área de trabalho são alterados para o momento do commit selecionado



Reflog

ENTENDENDO O COMANDO REFLOG

Acessar o repositório de Maria via Git Bash

Confirmar que a função validate não existe no arquivo login.js

```
$ cat login.js
```

Nota: esta função foi adicionada no commit que foi descartado na aula anterior

Utilizar o comando reflog para buscar a hash do commit que adicionou tal função

```
$ git reflog
```

Copiar a hash do commit

Alternar para o commit copiado

```
$ git switch <hash commit>
```

Nota: o Git informa um problema e sugere a utilização da opção `--detach`

Nota: este problema será explicado na aula de detached head, por hora simplesmente executar: `git switch <hash commit> --detach`

Verificar que a função retornou ao arquivo login.js

```
$ cat login.js
```



Detached HEAD

ENTENDENDO O ESTADO DETACHED HEAD

Acessar o repositório de Maria via Git Bash

Verificar o estado do repositório

```
$ git status
```

Nota: o Git informa "HEAD detached at XXX"

- Isso significa que a referência HEAD está desanexada
- A HEAD deve apontar para a branch atual
- É através da referência HEAD que o Git identifica a branch corrente
- O comportamento normal é o HEAD sempre referenciar uma branch e a branch ser uma referência para um commit
- Ao fazer um switch diretamente em um commit, o HEAD irá referenciar diretamente este commit
- Essa situação caracteriza um detached HEAD

SAINDO DO ESTADO DETACHED HEAD

Criar uma nova branch a partir do commit atual

```
$ git branch temp
```

Retornar para a branch login

```
$ git switch login
```

Realizar um merge da temp em direção a branch login

```
$ git merge temp
```

Nota: merge é um tópico que será abordado melhor na próxima seção



SEÇÃO 4: DESCARTANDO ALTERAÇÕES

Remover a branch temporária

```
$ git branch -d temp
```

Verificar o estado do repositório

```
$ git status
```

Nota: o Git não informa mais "HEAD detached at XXX"

Verificar que a função validate ainda permanece no arquivo login.js

```
$ cat login.js
```

Verificar o histórico da branch atual

```
$ git hist
```

Nota: a HEAD novamente é uma referência para a branch login, que é neste momento a branch atual



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.**

