

# Exemplos de Rotina

*PET Estatística UFPR*



# Sumário

A	Exemplos de Rotinas
---	---------------------

5
---



# Apêndice A

## Exemplos de Rotinas

Neste apêndice são descritas brevemente algumas das rotinas mais usuais em projetos Git. A maioria dos exemplos estão detalhados na apostila e foram resumidos nesta coletânea. Abaixo temos a lista de rotinas aqui presentes:

Configure usuário e e-mail, <a href="#">pág. 6</a>	Volte versões anteriores, <a href="#">pág. 11</a>
Inicie um projeto git local, <a href="#">pág. 6</a>	Reescreva <i>commits</i> , <a href="#">pág. 12</a>
Trabalhe com ramos, <a href="#">pág. 7</a>	Adicione chaves públicas, <a href="#">pág. 13</a>
Visualize diferenças, <a href="#">pág. 8</a>	Trabalhe remotamente, <a href="#">pág. 13</a>
Resolva conflitos de <i>merge</i> , <a href="#">pág. 9</a>	Manipule ramos remotos, <a href="#">pág. 14</a>
Visualize seu projeto, <a href="#">pág. 10</a>	Adicione local remoto, <a href="#">pág. 16</a>

## Configure usuário e e-mail

---

```
## Configurando localmente
## - válido para o repositório atual
git config user.name "Name Lastname"
git config user.email "namelastname@servidor"

## Configurando globalmente
## - válido para todos os repositórios do computador
git config --global user.name "Name Lastname"
git config --global user.email "namelastname@servidor"

## Obs.: As configurações locais se sobrepõem as
## globais, quando as duas forem avaliadas.
```

---

## Inicie um projeto Git local

---

```
## Em um diretório que deseja-se versionar

## Inicia o versionamento Git
git init
```

```
## Verifica o estado do repositório  
git status  
  
## Adicione os arquivos para receber o commit  
git add file1.txt file2.R file3.txt file4.Rmd  
  
## Registre a versão do repositório com uma mensagem  
## informativa  
git commit -m "Inicia repositório"  
  
## Verifica o histórico de versões  
git log
```

---

## Trabalhe com ramos

---

```
## Verifica os ramos existentes  
git branch  
  
## Cria um ramo, para trabalho específico como:  
## - Corrigir bugs  
## - Desenvolver features  
## - Tarefas em geral  
git branch bugfix  
  
## Altera versionamento para o ramo criado
```

```
git checkout bugfix

## Adiciona as alterações que corrigem o bug
git add app.R

## Registra as alterações
git commit -m "Altera delimitadores da função"

## Retorna para o ramo principal
git checkout master

## Incorpora ao ramo principal as alterações
## realizados no ramo bugfix
git merge bugfix

## Deleta o ramo responsável pela correção do bug
git branch -d bugfix
```

---

## Visualize diferenças

---

```
## -----
## Diferenças não commitadas

## Lista as diferenças entre o último commit e o
## estado do repositório no working directory
```



```
git diff

## Diferença para um único arquivo
git diff file.Rmd

## -----
## Diferenças entre versões commitadas

## Verifica os registros com referência
git reflog

## Lista as diferenças entre o último e o
## antepenúltimo registro
git diff HEAD~0 HEAD~2 ## ou

## -----
## Diferenças entre ramos

## Lista diferenças entre os ramos master e
## feature1
git diff master feature1

## Obs.: Programas externos podem ser utilizados para
## visualizar diferenças use difftool no lugar de
## diff, ver no cap. 06
```

## Resolva conflitos de merge

---

```
## Incorpora modificações realizadas no ramo feature  
git merge feature  
  
## Edite e salve o(s) arquivo(s) em conflito, as  
## porções em conflito são destacadas por <<< === >>>  
  
## Finaliza merge com o commit:  
git commit -a -m "Merge do ramo feature"
```

---

## Visualize seu projeto

---

```
## Histórico de registros  
git log  
  
## Histórico de registros em uma linha com trajetória  
## dos ramos  
git log --graph --oneline  
  
## Histórico de registro com referência  
git reflog
```

```
## Histórico de registro via interface gráfica padrão  
gitk
```

---

## Volte versões anteriores

---

```
## Verifica o histórico de versões do repositório  
## - Guarde o SHA1 ID do registro desejado: ec3650c8  
git log --oneline  
  
## -----  
## Descartar todas as alterações até um commit  
git reset --hard ec3650c8  
  
## Obs.: Reescreve a história do repositório, não  
## é recomendável reescrever a linha do tempo quando  
## se está em um projeto colaborativo remoto.  
  
## -----  
## Reverte o estado atual para um registro específico  
git revert ec3650c8  
git commit -am "Retorna projeto para versão funcional"  
  
## Obs.: Faz um merge da versão atual com a versão do  
## SHA1 ID informado. P0dem haver conflitos que devem
```

```
## ser resolvido para concluir a reversão.

## -----
## Cria um ramo provisório a partir de um SHA1 ID
git checkout ec3650c8

## Visualiza os ramos existentes
git branch

## Cria um ramo definitivo com o estado no SHA1 ID
git checkout -b ramo_teste

## Obs.: O ramo provisório é removido assim que se
## fizer um checkout para qualquer outro ramo
```

---

## Reescreva commits

---

```
## Verifica o histórico de versões do repositório
git log --oneline

## -----
## Reescreve a última mensagem de commit
git commit --amend -m "Correção de Commit"

## Obs1.: Arquivos na staging area também são
```

```
## incorporados ao último commit
## Obs2.: Reescreve a história do repositório, não
## é recomendável reescrever a linha do tempo quando
## se está em um projeto colaborativo remoto.
```

---

## Adicione chaves públicas

---

```
## Crie uma chave pública.
ssh-keygen -t rsa -C "namelastname@servidor"

## Exibe as chaves públicas.
cat ~/.ssh/id_rsa.pub

## Adicione o conteúdo a um servidor remoto, como:
## - git@github.com
## - git@gitlab.com
## - git@gitlab.c3sl.ufpr.br

## Verifica conexão com o servidor
ssh -T endereço ## endereço = git@github.com, ...

## Obs.: Todos os comandos ssh são provenientes do
## pacote de função ssh para shell, portanto para
## utilizar instale este pacote.
```

## Trabalhe remotamente

---

```
## Clona um projeto remoto:  
## e.g. git@github.com:pet-estatistica/apostila-git.git  
git clone endereço:namespace/project.git  
  
## Realiza modificações e/ou inclusões de em um ou  
## vários arquivos  
  
## Adiciona todas as alterações para commit  
git add .  
  
## Registra suas alterações  
git commit -a -m "Modifica compilação do projeto"  
  
## Envia as alterações para o repositório remoto (origin)  
git push origin  
  
## Traz estado do repositório remoto  
git pull origin
```

---

## Manipule ramos remotos

---

```
## -----  
## Realizando contribuições em um ramo remoto  
  
## Lista todos os ramos  
git branch -a  
  
## Altera versionamento para o ramo issue01  
git checkout issue01  
  
## Traz estado do ramo remoto issue01  
git pull origin issue01  
  
## Realiza modificações e/ou inclusões de em um ou  
## vários arquivos  
  
## Adiciona todas as alterações para commit  
git add .  
  
## Registra suas alterações  
git commit -a -m "Modifica laço iterativo condicional"  
  
## Envia as alterações no ramo para a versão remota  
git push origin issue01  
  
## -----  
## Realizando contribuições em um ramo remoto e  
## enviando o ramo principal mesclado
```

```
## Traz estado do ramo remoto issue01
git pull origin bugfix

## Altera versionamento para o ramo issue01
git checkout bugfix

## Realiza modificações e/ou inclusões de em um ou
## vários arquivos

## Adiciona todas as alterações para commit
git add .

## Registra suas alterações
git commit -a -m "Altera classe do objeto retornado"

## Retorna ao ramo principal
git checkout master

## Incorpora modificações realizadas no ramo bugfix
git merge bugfix

## Envia as alterações o repositório remoto
git push origin master

## Deleta o ramo local e remoto bugfix
git branch -d bugfix ## Local
git push :bugfix      ## Remoto
```



---

## Adicione local remoto

---

```
## Lista os servidores remotos, com endereço
git remote -v

## -----
## Adicionando local para trazer contribuições

## Adiciona local remoto com nome gitlab:
git remote add gitlab git@gitlab.com:user/project.git

## Adiciona local remoto com nome github:
git remote add github git@github.com:user/project.git

## Atualiza arquivos locais, baseado no local remoto:
git pull gitlab ## gitlab.com:namespace/project.git
git pull github ## github.com:namespace/project.git

## -----
## Adicionando local para enviar contribuições

## Lista os locais de origem
git remote show origin

## Adiciona novo local de origem:
## e.g. git@github.com:pet-estatistica/apostila-git.git
```

```
git remote set-url origin --push --add endereço_remoto  
## Envia as contribuições para os locais remotos  
git push origin
```