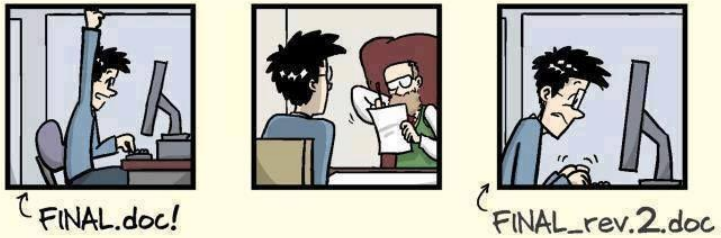
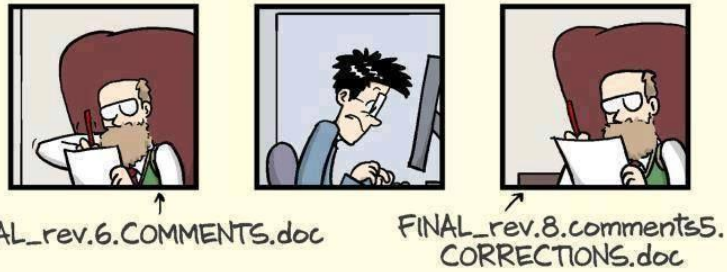
DEPARTAMENTO DE FÍSICA NUCLEAR E ALTAS ENERGIAS

Git/GitHub tutorial

Introdução à análise de dados em FAE

"FINAL".doc





O que é o Git?

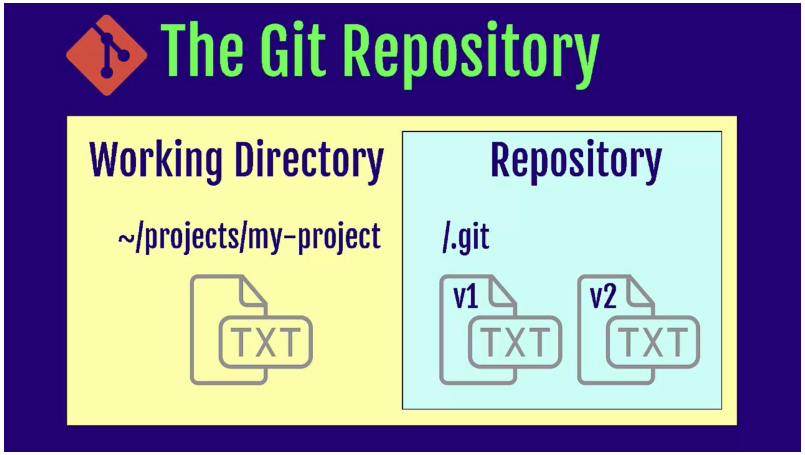
- Git é um sistema de controle de versão de arquivos
- Criado por Linus Torvalds (2005, criador do Linux)
- Para que serve?
  - Podemos sincronizar com repositórios online: baixar códigos de terceiros ou da sua equipe, e também enviar alterações.
  - Podemos ver as versões anteriores e visualizar o que mudou de uma versão para outra.

Repositórios

- Um repositório do Git é um armazenamento virtual para projetos.
- Ele permite que você salve versões do código, que você pode acessar quando precisar.
- O repositório pode conter:
  - Códigos fontes
  - Arquivos de texto README (o GitHub usa o formato Markdown)
  - Arquivos de entrada (de vários tipos)
  - `.gitignore`: lista de tipos de arquivos (ou nome dos arquivos) que devem ser ignorados automaticamente
- **Não** coloque arquivos binários grandes (por exemplo, arquivos .root) em um repositório Git! Nunca!
  - O tamanho do repositório pode explodir
  - As operações podem ficar muito lentas

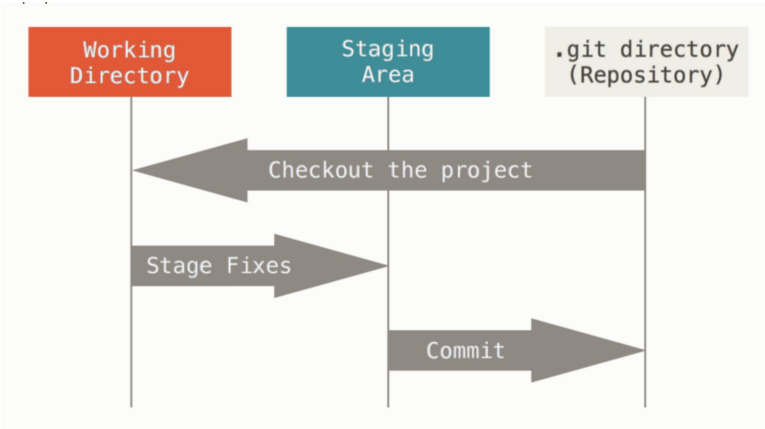
Repositório Git

- Um repositório Git **é um diretório** (pasta) no computador que **registra as alterações feitas nos arquivos** que estão contidas nele e adicionados ao índice do repositório.
- Dentro do repositório temos uma pasta oculta chamada ***.git***, e nela contém os arquivos de configuração do git.



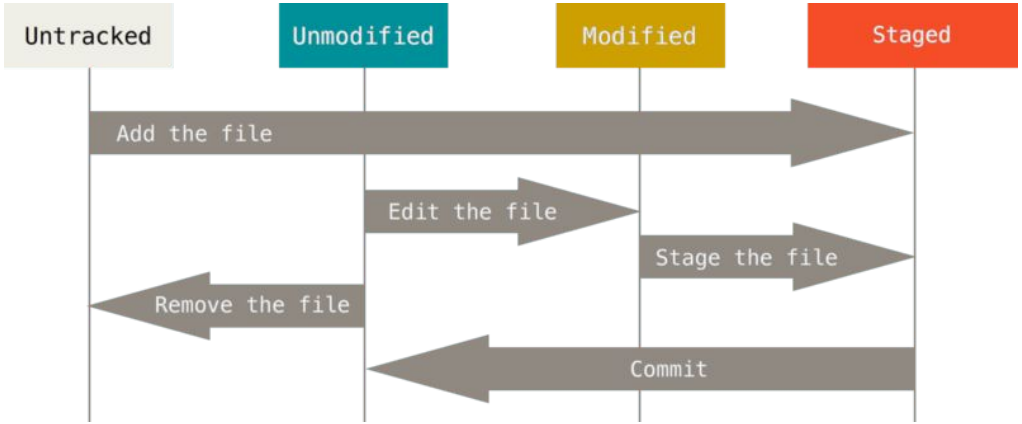
Repositório Git

- O Git tem 3 estados principais que seus arquivos podem estar:
  - Commited: significa que os dados estão armazenados de forma segura em seu banco de dados local.
  - Modificado (modified): significa que o arquivo foi alterado mas ainda não fez o *commit* no seu banco de dados.
  - Preparado (staged): significa que a versão atual do arquivo de um arquivo modificado foi marcado para fazer parte do próximo *commit*.



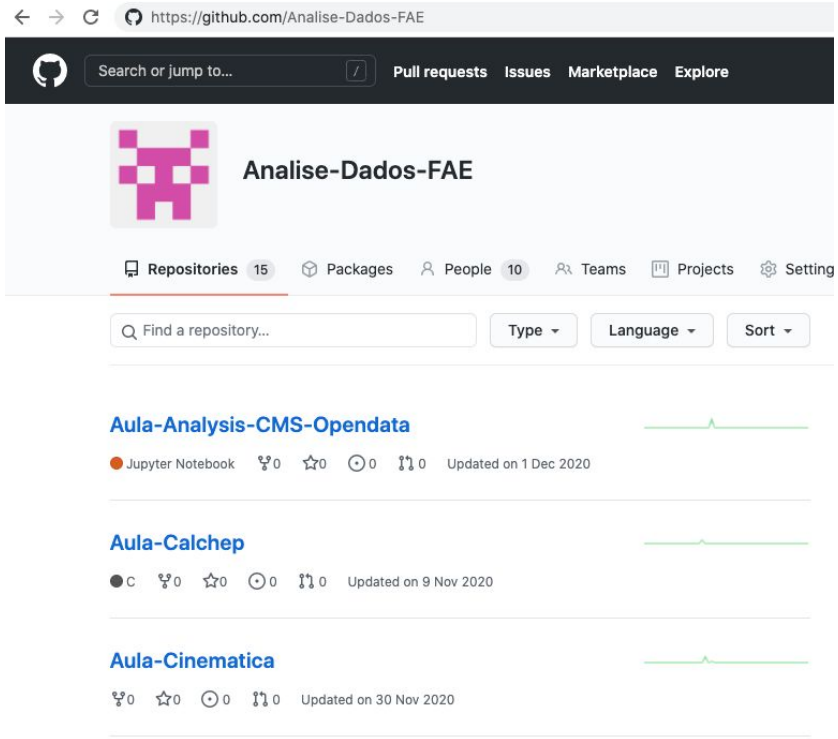
Fundamentos do Git

- Cada arquivo em seu diretório de trabalho pode estar em dos seguintes estados:
  - Rastreados: os arquivos rastreados foram incluídos no último *commit*; podem ser não modificados, modificados ou preparados.
  - Não-rastreados: são todos os arquivos que não foram incluídos no último *commit* e não estão na área de *stage*,



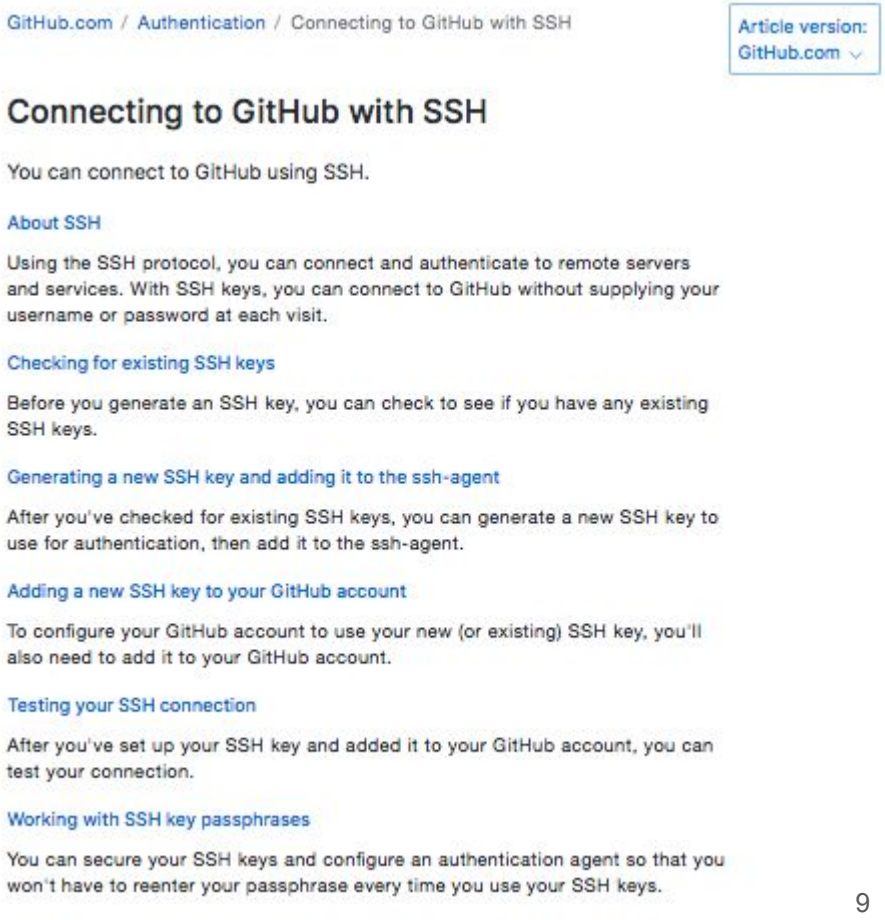
O que é o GitHub?

- É uma plataforma para hospedagem dos projetos.
- Diferença entre Git e GitHub:
  - Git é a ferramenta para versionar os projetos
  - Git mantém o histórico
  - O Git é usado para registrar as alterações do projeto
  - GitHub é o lugar que os projetos ficam
  - Existem outras alternativas para o GitHub, como por exemplo GitLab e BitBucket.



Criando e Configurando uma conta no GitHub

- Crie uma conta: https://github.com/join
- Crie uma chave SSH e adicione à sua conta: https://help.github.com/articles/generating-an-ssh-key/



GitHub.com / Authentication / Connecting to GitHub with SSH

Article version: GitHub.com

Connecting to GitHub with SSH

You can connect to GitHub using SSH.

About SSH

Using the SSH protocol, you can connect and authenticate to remote servers and services. With SSH keys, you can connect to GitHub without supplying your username or password at each visit.

Checking for existing SSH keys

Before you generate an SSH key, you can check to see if you have any existing SSH keys.

Generating a new SSH key and adding it to the ssh-agent

After you've checked for existing SSH keys, you can generate a new SSH key to use for authentication, then add it to the ssh-agent.

Adding a new SSH key to your GitHub account

To configure your GitHub account to use your new (or existing) SSH key, you'll also need to add it to your GitHub account.

Testing your SSH connection

After you've set up your SSH key and added it to your GitHub account, you can test your connection.

Working with SSH key passphrases

You can secure your SSH keys and configure an authentication agent so that you won't have to reenter your passphrase every time you use your SSH keys.

Configurando o Git: `git config`

- O Git armazena as opções de configuração em 3 arquivos separados, o que permite que você estenda as opções a repositórios individuais (local), usuário (Global) ou o sistema inteiro (sistema):
  - Local: `/.git/config`: Configurações específicas dos repositórios. Para configurar localmente adicione a opção `--local` ou não adicione nenhuma opção de configuração para o repositório local.
  - Global: `/.gitconfig`: Configurações específicas do usuário. É aqui que as opções definidas com o sinalizador `--global` são armazenadas.
  - Sistema: `$(prefix)/etc/gitconfig`: Configurações de todo o sistema.

Comando para configurar:

```
git config --global user.name [Name]
git config --global user.email [Email]
git config --global user.github [Account]
```

Para listar as configurações:

```
git config --list
```

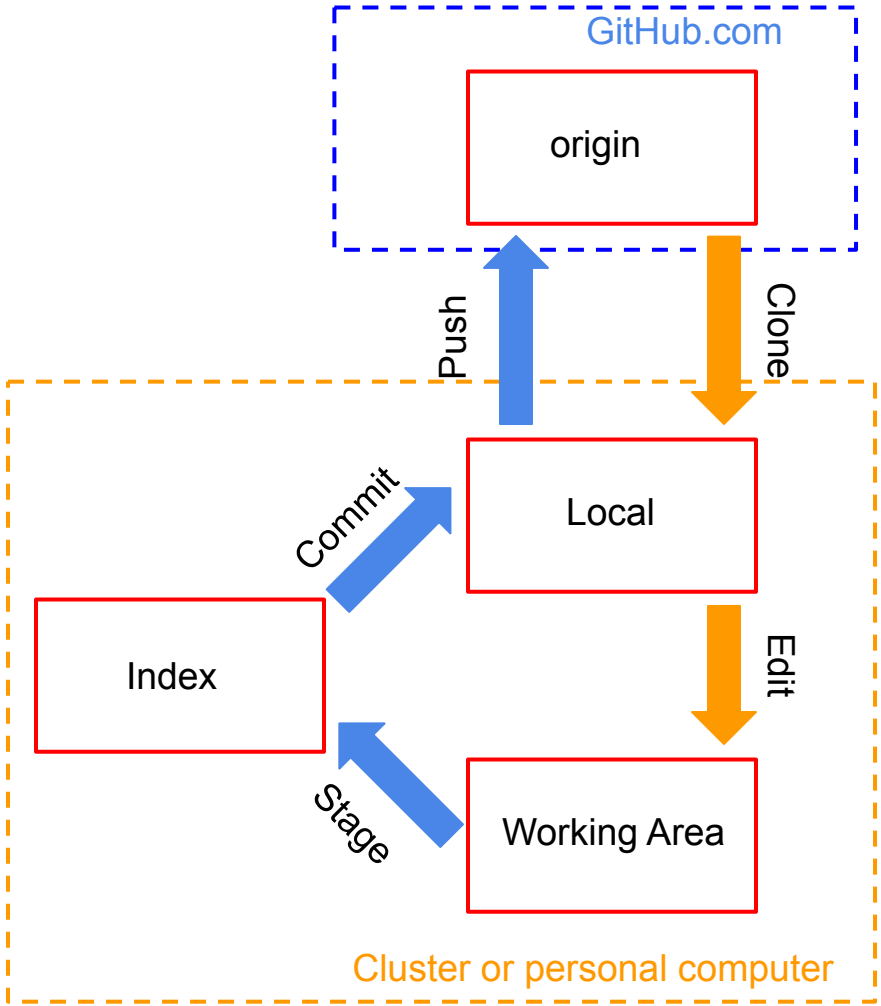
Para ver o arquivo de configurção global:

```
cat ~/.gitconfig
```

```
[sheilaamaral@iMacs-iMac analise-dados-FAE]$ git config --global user.name "Sheila Amaral"
[sheilaamaral@iMacs-iMac analise-dados-FAE]$ git config --global user.email "sheila.silva.do.amaral@cern.ch"
```

Repositório remoto

- Área de trabalho: local onde está o seu código localmente
- Área de preparo: área temporária onde ficam as alterações feitas na sua área de trabalho
- Local: diretório .git na sua máquina
- Origin: repositório no servidor remoto



Termos comuns

- **Branch (ramificação)**: área de desenvolvimento do projeto
  - Um repositório pode ter vários branchs
- **Commit**: são os registros das alterações que foram realizadas num projeto (modificar, adicionar, remover arquivos)
  - Um commit é simplesmente um *diff* (lista das linhas alterados, adicionadas, removidas)
  - Cada commit é identificado por um *hash* (SHA-1) baseado em metadados: *parent* (commit anterior), data/hora, autor, committer, mensagem, etc.
- **Tag**: versão específica do projeto
- **Tracked**: arquivos que estão sendo controlados pelo Git
- **Untracked**: arquivos que não estão sendo controlados pelo Git

Comandos básicos: como criar um repositório

Como configurar um repositório sob o controle de versão do Git:

- **`git init`**: inicia um novo repositório do git localmente. Esse comando cria um novo subdiretório .git no diretório de trabalho atual. Isso também cria um novo branch principal.
- **`git clone https://github.com/[user]/[repo].git`**
  - Copia um novo repositório diretamente do GitHub
  - Caso queira copiar o repositório [repo] e salvar com o nome my-repo: **`git clone https://github.com/[user]/[repo].git my-repo`**
  - Para copiar de um branch específico: **`git clone https://github.com/[user]/[repo].git -b new`**

Comandos básicos: salvando alterações

- **`git add arquivo`**: para começar o rastreamento do arquivo pelo Git, ou todos os arquivos (.). Após a execução desse comando, o arquivo está sendo rastreado e preparado para o *commit*.
- **`git commit -a -m "nova versão"`**: define que as alterações até aquele momento são uma versão do código. Grava as alterações do seu projeto que estavam na área do stage.
- **`git push origin master`**: envia os arquivos do último *commit* ao repositório online.
- **`git pull origin master`**: baixa o último commit de um repositório online.

Comandos básicos: operações com arquivos

- **`git rm [file]`**: deleta um arquivo e para de rastreá-lo
- **`git mv [old] [new]`**: renomeia um arquivo

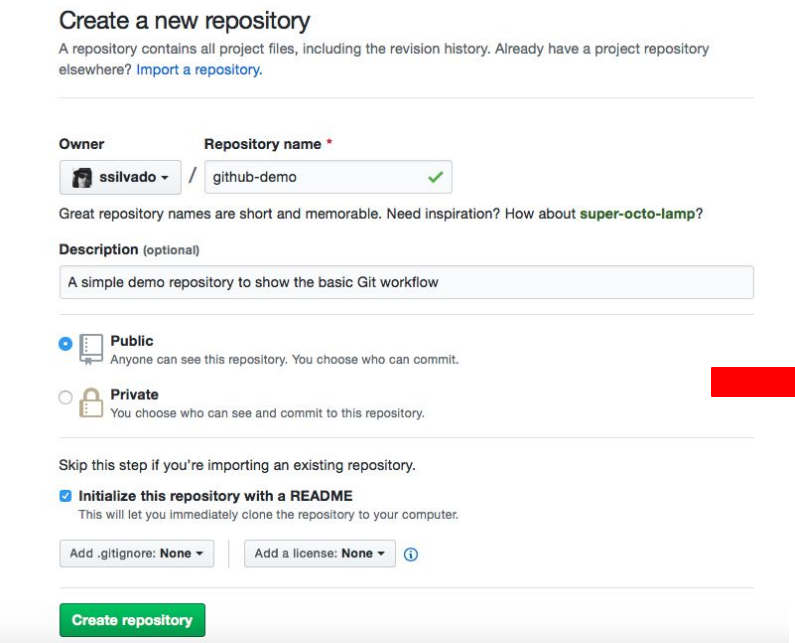
Comandos básicos: Branches

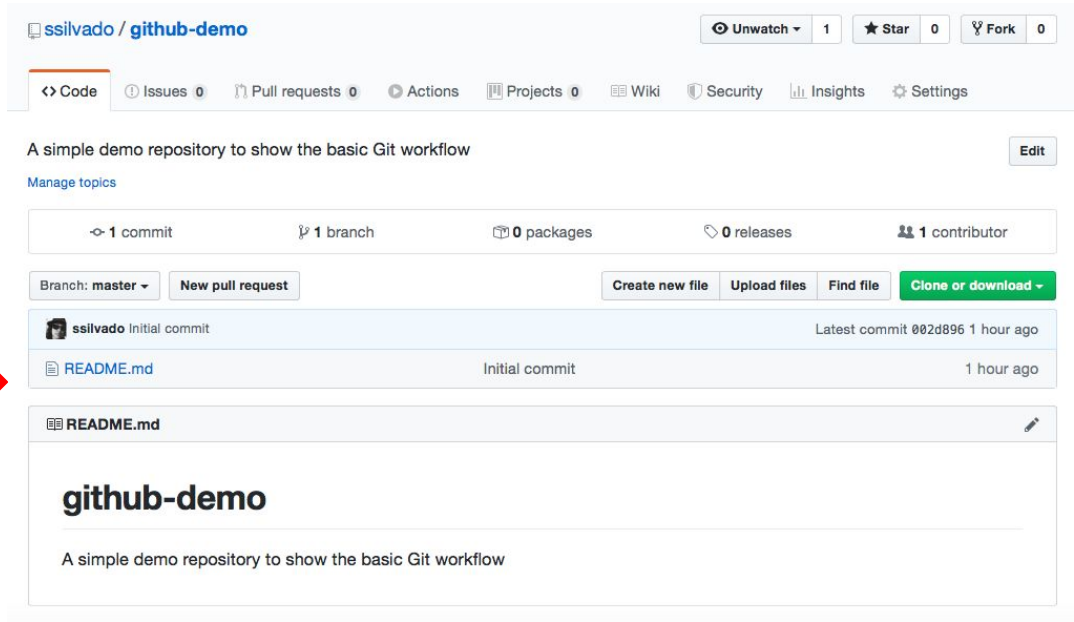
- **`git checkout [branch]`**: para mudar para uma branch existente
- **`git checkout -b [branch]`**: criar um novo branch e te redireciona para ele
- **`git branch`**: lista todas as branches locais
  - `git branch -a`: listar todas as *branchs*, inclusive as remotas
  - `git branch [new]` : criar uma nova branch
  - `git branch -m [old] [new]` : renomear uma branch
  - `git branch -d [branch]` : deletar uma branch

Basic Commands: Status

- **`git status`**: verifica o status dos seus arquivos. Por exemplo, se existem novos arquivos que precisam ser adicionados, se existem alterações, se arquivos foram excluídos e se o diretório é um repositório git.
  - Este comando ignora os arquivos listado em `.gitignore`
  - `-s`: status resumido
  - `-suno`: status resumido (`s`) dos arquivos rastreados (`uno`)
  - `[file]`: status de um arquivos específico
- **`git diff`**: ver as alterações no formato *diff*
  - Várias opções: arquivos específico (`[file]`), commits específicos (`[commit1] [commit2]`), etc.

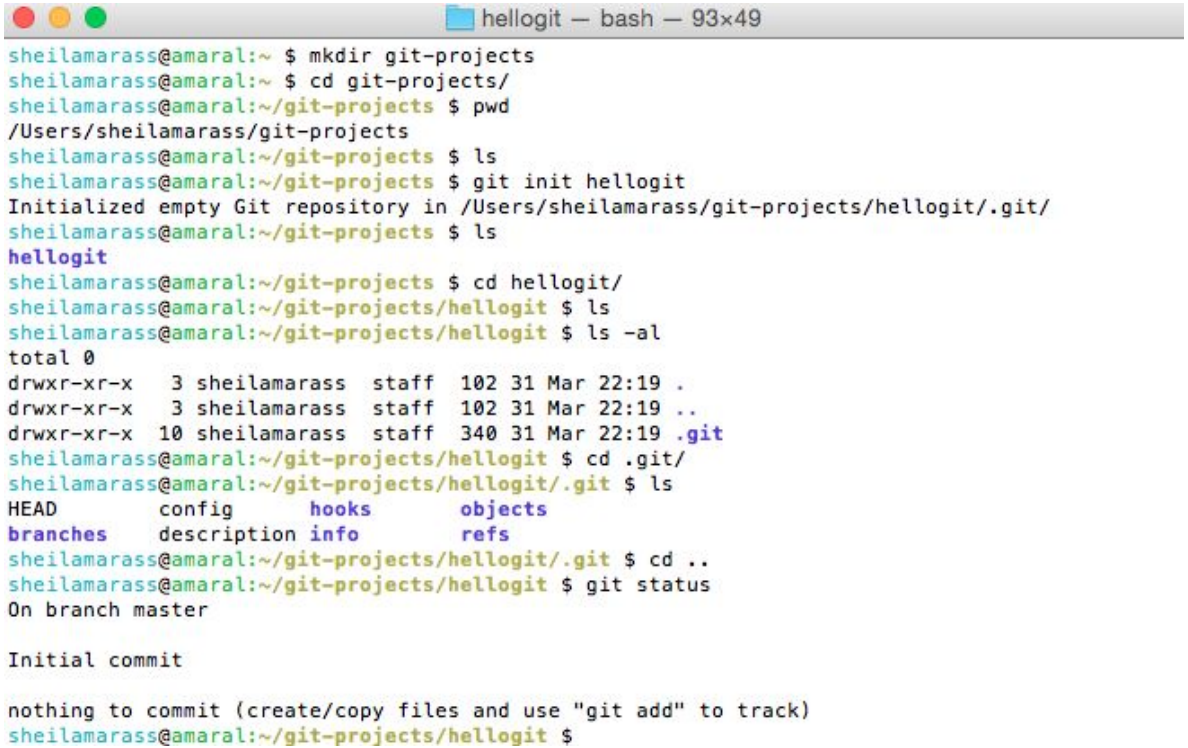
Como criar um repositório no GitHub





Como criar um repositório localmente

https://hipsum.co/



Criando um repositório:

```
cd git-projects
git init hellogit
```

Checando o diretório .git:

```
cd hellogit
ls .git
```

Checando o status:

```
git status
```

Como criar um repositório localmente - continuação

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

Criando um novo arquivo test.txt:

```
echo "It is just a demo file" >> test.txt
```

Checando o status:

```
git status
```

Inicializar o rastreamento do arquivo test.txt:

```
git add test.txt
git status
```

Enviamos as alterações para a área de stage:

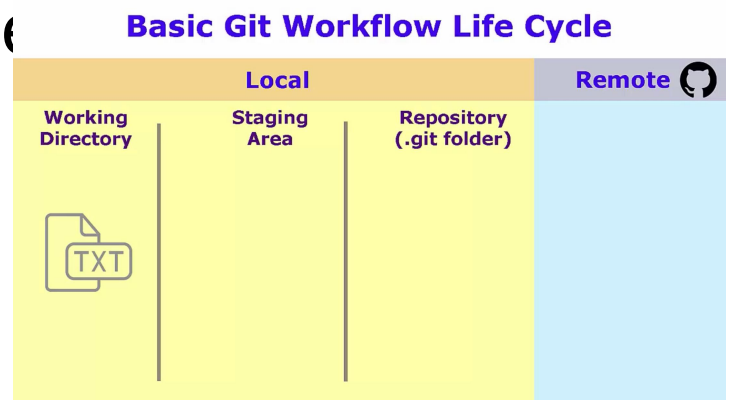
```
git commit
git status
```

Checando os registos:

```
git log
```

Como criar um repositório localme

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```



Inicializar o rastreamento do arquivo test.txt:

```
git add test.txt
git status
```

Enviamos as alterações para a área de stage:

```
git commit
git status
```

Checando os registos:

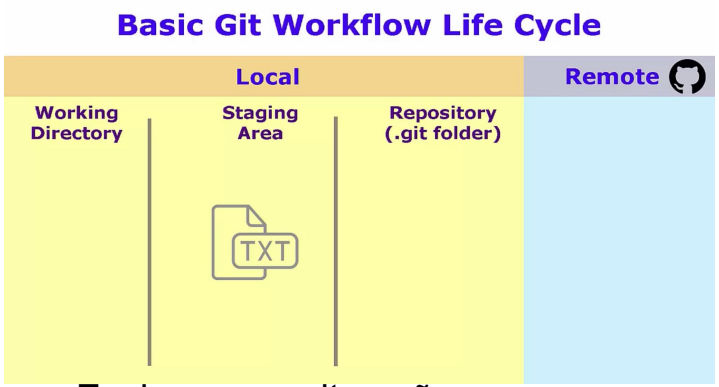
```
git log
```

Como criar um repositório localmente - continuação

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

Criando um novo arquivo test.txt:

```
echo "It is just a demo
file" >> test.txt
```



Enviamos as alterações para a área de stage:

```
git commit
git status
```

Checando os registos:

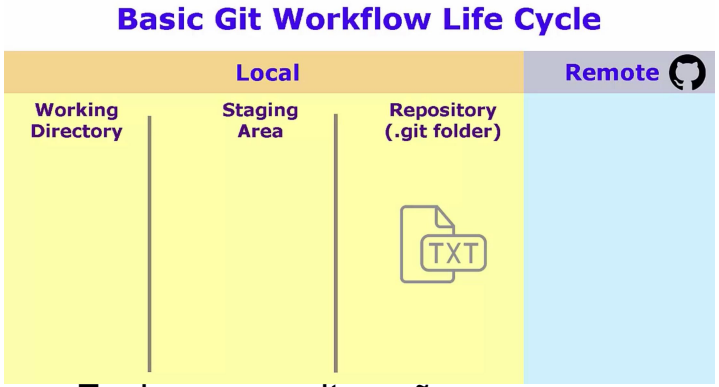
```
git log
```

Como criar um repositório localmente - continuação

```
sheilamarass@amaral:~/git-projects/hellogit $ echo "It is just a demo file" >> test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
sheilamarass@amaral:~/git-projects/hellogit $ git add test.txt
sheilamarass@amaral:~/git-projects/hellogit $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   test.txt

sheilamarass@amaral:~/git-projects/hellogit $ git commit -m "A simple commit"
[master (root-commit) 1f67f82] A simple commit
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
sheilamarass@amaral:~/git-projects/hellogit (master) $ git status
On branch master
nothing to commit, working directory clean
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

Criando um novo arquivo test.txt:

```
echo "It is just a demo
file" >> test.txt
```



Enviamos as alterações para a área de stage:

```
git commit
git status
```

Checando os registos:

```
git log
```

Criando um repositório no GitHub

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner **Repository name** *

ssilvado / hellogit

Great repository names are short and memorable. Need inspiration? How about **sturdy-telegram**?

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

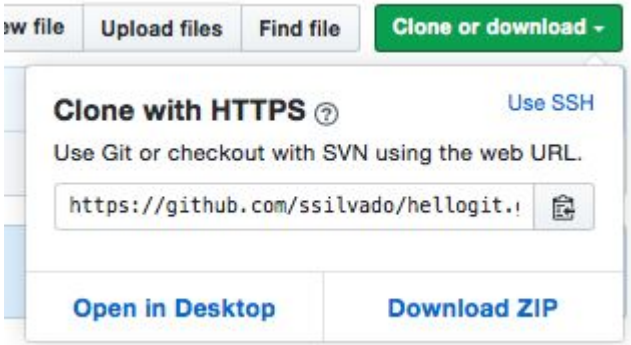
Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** Add a license: **None**

Create repository



Vinculando um repositório local com o GitHub - `git push`

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote add origin
https://github.com/ssilvado/hellogit.git
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
origin  https://github.com/ssilvado/hellogit.git (fetch)
origin  https://github.com/ssilvado/hellogit.git (push)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ssilvado/hellogit.git
 * [new branch]      master -> master
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

Mostrar a lista de repositórios remotos

`git remote -v`

Adicionar um repositório remoto

`git remote add origin https://github.com/ssilvado/hellogit.git`

Mostrar a lista de repositórios remotos

`git remote -v`

Enviar o repositório local ref (master) para um repositório remoto (origin):

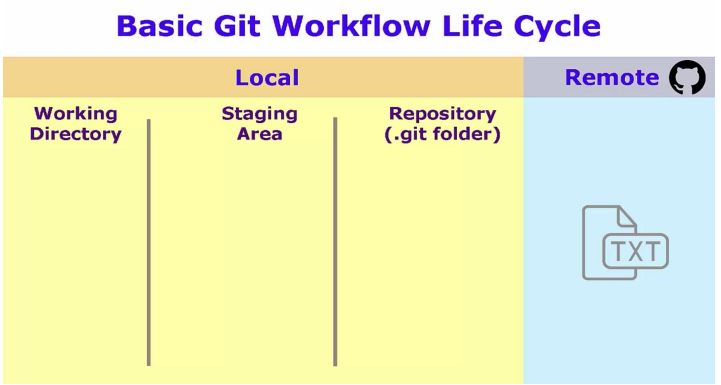
`git push origin master`

Vinculando um repositório local com o GitHub - `git push`

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote add origin
https://github.com/ssilvado/hellogit.git
sheilamarass@amaral:~/git-projects/hellogit (master) $ git remote -v
origin  https://github.com/ssilvado/hellogit.git (fetch)
origin  https://github.com/ssilvado/hellogit.git (push)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 244 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ssilvado/hellogit.git
 * [new branch]      master -> master
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

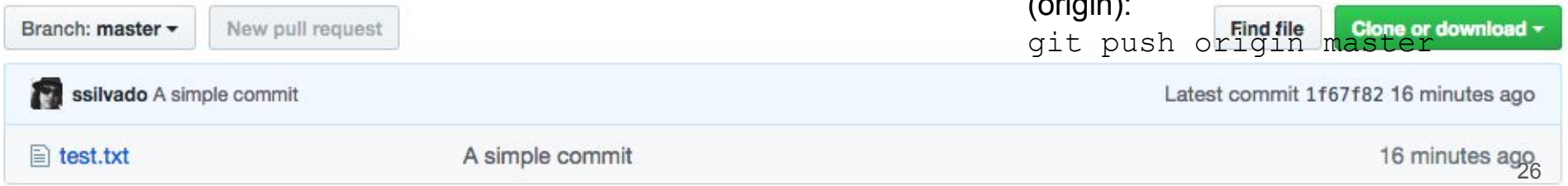
Mostrar a lista de repositórios remotos

`git remote -v`



Enviar o repositório local ref (master) para um repositório remoto (origin):

`git push origin master`



Criando um novo branch

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch -v
* master 1f67f82 A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch work
sheilamarass@amaral:~/git-projects/hellogit (master) $ git checkout work
Switched to branch 'work'
sheilamarass@amaral:~/git-projects/hellogit (work) $ git log --all
commit 1f67f82ff927fb82fd8050b1059527929edbe447
Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
Date:   Tue Mar 31 22:21:53 2020 -0400

    A simple commit
```

Checar em qual branch estamos:
`git branch -v`

Criando um novo branch chamado work:
`git branch work`

Mudando para o branch work:
`git checkout work`

Exibir todo o histórico de alterações de todos os branches:
`git log --all`

Mesclando projetos - `git merge`

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git checkout -b testing master
Switched to a new branch 'testing'
sheilamarass@amaral:~/git-projects/hellogit (testing) $ echo "Working with branches and merge" >>
test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git add test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git commit -m "New test"
[testing d144418] New test
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git checkout master
Switched to branch 'master'
sheilamarass@amaral:~/git-projects/hellogit (master) $ git merge testing
Updating c37981c..d144418
Fast-forward
 test-branches-merge.txt | 1 +
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch -d testing
Deleted branch testing (was d144418).
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log --graph
* commit d14441841907e1e6de826c163f1be14edb4aea25
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:14:42 2020 -0400
|
|     New test
|
* commit c37981c5ad4899824fa428322e400c8900758df5
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:10:17 2020 -0400
|
|     Start feature
|
* commit 1f67f82ff927fb82fd8050b1059527929edbe447
  Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
  Date:   Tue Mar 31 22:21:53 2020 -0400

      A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```

Criamos um novo branch:

```
git checkout -b testing master
```

Criamos um arquivo test-branches-merge.txt e fizemos o commit

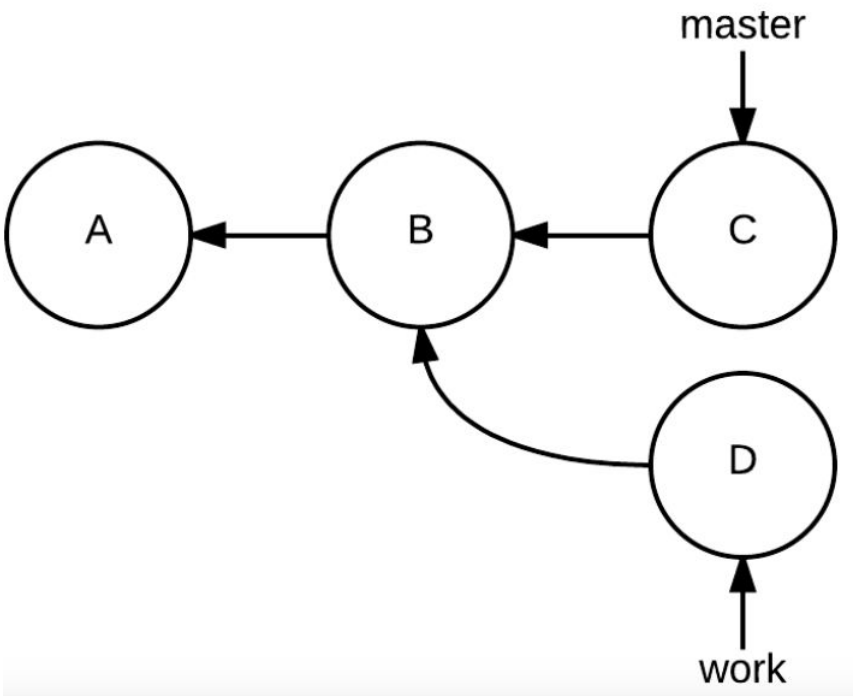
```
echo “Working with branches
and merge” >>
test-branches-merge.txt
git add
test-branches-merge.txt
git commit -m “New test”
```

Mesclamos a alteração na branch testing:

```
git checkout master
git merge testing
git branch -d testing
```

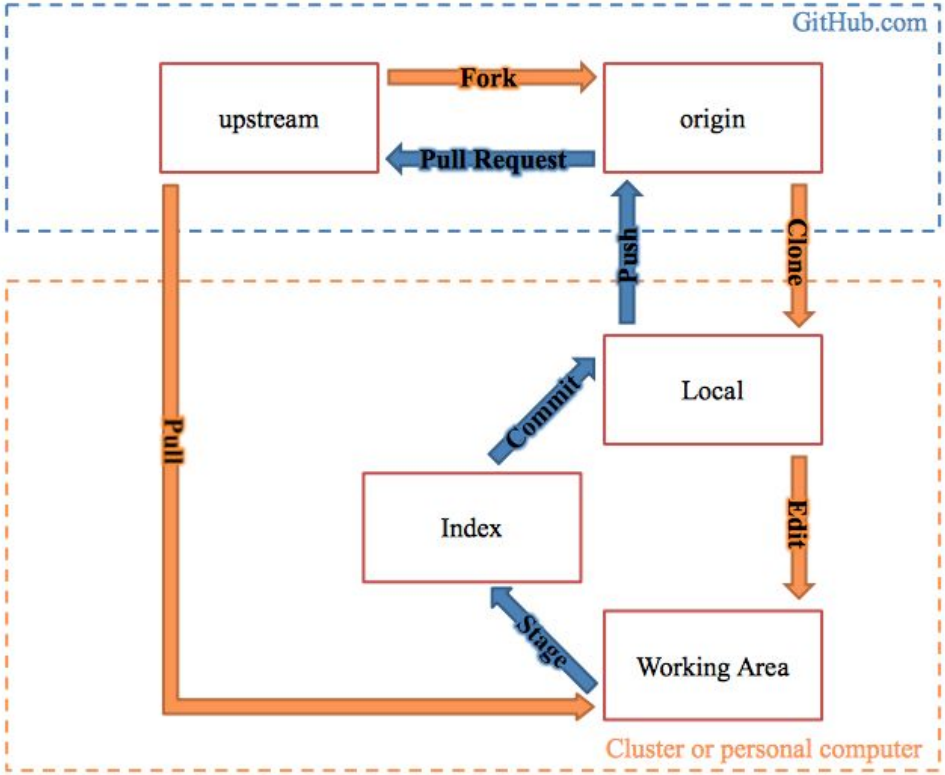
Mesclando projetos - `git merge`

```
sheilamarass@amaral:~/git-projects/hellogit (master) $ git checkout -b testing master
Switched to a new branch 'testing'
sheilamarass@amaral:~/git-projects/hellogit (testing) $ echo "Working with branches and merge" >>
test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git add test-branches-merge.txt
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git commit -m "New test"
[testing d144418] New test
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (testing) $ git checkout master
Switched to branch 'master'
sheilamarass@amaral:~/git-projects/hellogit (master) $ git merge testing
Updating c37981c..d144418
Fast-forward
 test-branches-merge.txt | 1 +
 1 file changed, 1 insertion(+)
sheilamarass@amaral:~/git-projects/hellogit (master) $ git branch -d testing
Deleted branch testing (was d144418).
sheilamarass@amaral:~/git-projects/hellogit (master) $ git log --graph
* commit d14441841907e1e6de826c163f1be14edb4aea25
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:14:42 2020 -0400
|
|     New test
|
* commit c37981c5ad4899824fa428322e400c8900758df5
| Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
| Date:   Tue Mar 31 23:10:17 2020 -0400
|
|     Start feature
|
* commit 1f67f82ff927fb82fd8050b1059527929edbe447
  Author: Sheila Amaral <sheila.silva.do.amaral@cern.ch>
  Date:   Tue Mar 31 22:21:53 2020 -0400

      A simple commit
sheilamarass@amaral:~/git-projects/hellogit (master) $
```



Colaboração no GitHub - Pull Request

- Efetuar um Pull Request significa informa outros que você irá implementar as mudanças (patches) criadas no seu branch ao repositório master (upstream).
- Os colaboradores do repositório podem aceitar ou negar o PR.
- Muito utilizado entre colaboradores de pacotes/projetos:
  - Ao invés de todos fazerem commit para o mesmo repositório ✗
  - Cada contribuidor copia (forks) o repositório e envia um (pull requests) com o novo código ✓ ⇒ modelo “Fork and Pull”



Fluxo do “Fork and Pull”

1. Faça o fork do projeto: crie uma cópia de um repositório de uma projeto existente no GitHub de um usuário, para a sua conta
2. Clone o repositório para o seu computador local: `git clone https://github.com/useraccount/demo`
3. Crie uma branch para alterações no master: `git checkout -b new_branch`
4. Adicione o repositório remoto para receber as alterações: `git remote add upstream https://github.com/useraccount/demo`
   a. Neste caso, “upstream repo” se refere ao repositório original que você criou quando fez o fork.
5. Realize as alterações no código e faça os commits:
   a. Por exemplo: (`echo “some test file” >> test`) > `git status` > `git add test` > `git commit -m “Adding a test file to new_branch”`
6. Faça o push dessa branch para o seu projeto no GitHub: `git push -u origin new_branch`
7. Abra um Pull Request no GitHub

Desfazendo ações no Git

- Modificando seu último commit: `git commit --amend`
- Retirando um arquivo do stage (antes `git commit`): `git reset HEAD <file>`
- Desfazendo alterações em um arquivo modificado: `git checkout -- <file>`

Exercício

- Utilizando o método “Fork and Pull”, faça alguma alteração no projeto e submeta um Pull Request.
- O repositório é: https://github.com/ssilvado/web-project (*)

*Este projeto HTML5 foi criado usando http://www.initializr.com/, que é um gerador de projetos.