



# Curso PHP e MySQL



## \_Iniciando com controle de versão (git e github)

Introdução às tecnologias, instalação do GIT localmente, criando conta no Github, clonando e gerenciando repositórios remotos.

# GIT

É um sistema de controle de versão de arquivos.

Através deles podemos desenvolver projetos no qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.

<https://git-scm.com/>



# GITHUB

É uma plataforma de hospedagem de código-fonte com controle de versão usando o Git.

Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.

<https://github.com/>



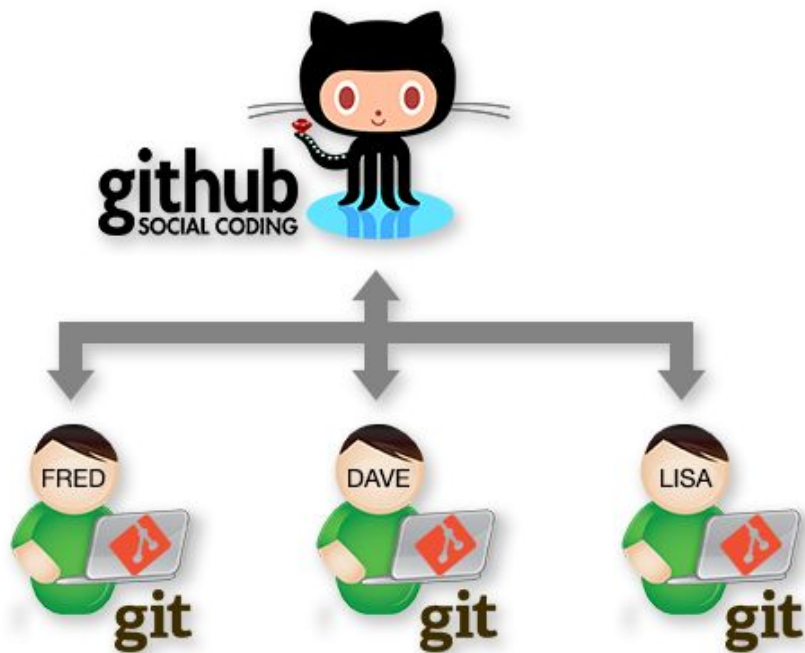
# Controle de Versão é diferente de Repositório



Trabalho  
**localmente**  
em sua máquina,  
gerenciando os  
seus repositórios.

Armazeno milhares de  
repositórios  
**remotamente,**  
para você acessar quando  
e onde quiser.

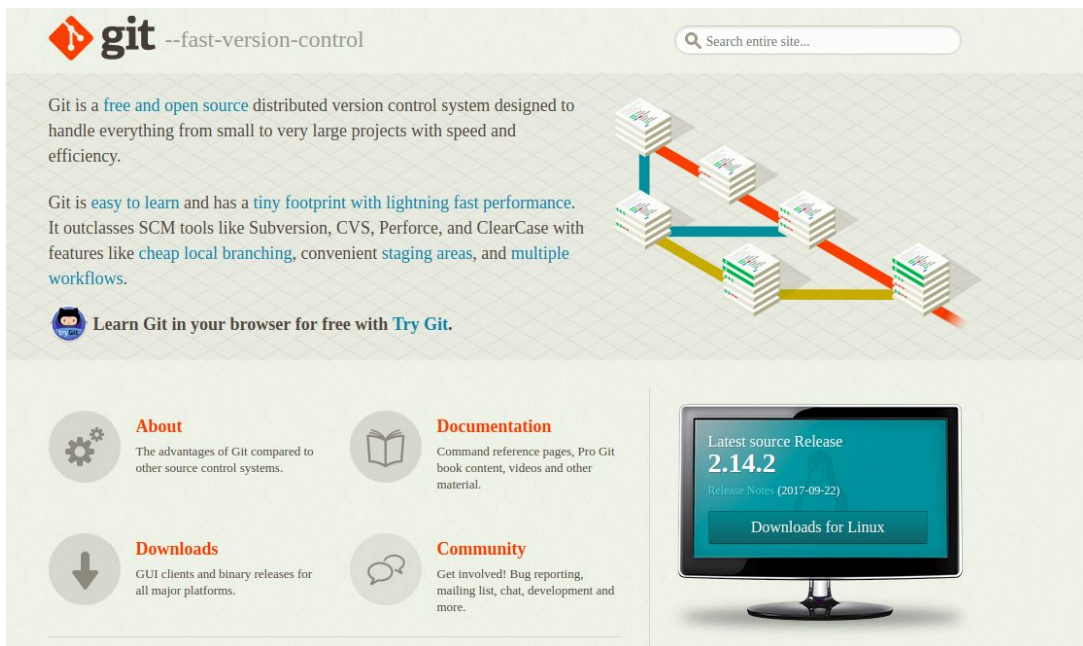
# COMO FUNCIONA?



Hoje o GitHub abriga mais de **65 milhões de projetos**, alguns deles que são conhecidos mundialmente. **WordPress, GNU/Linux e o Atom.**

# Passo 1:

Baixar o git e instalar em sua máquina: <https://git-scm.com/>



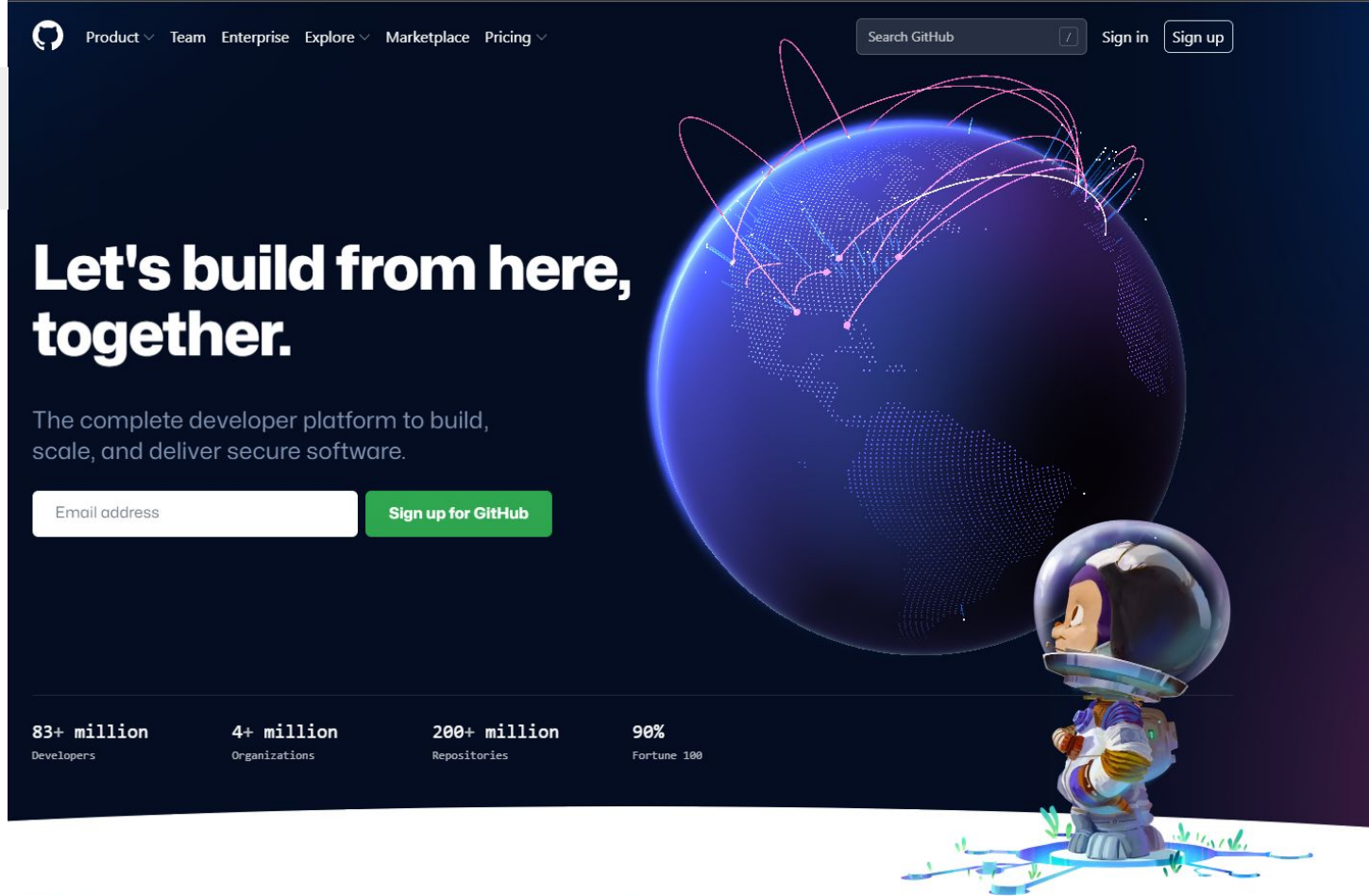
The screenshot shows the Git website homepage. At the top left is the Git logo with the tagline "--fast-version-control". To the right is a search bar. The main content area has two paragraphs: "Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency." and "Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows." Below this is a GitHub logo and the text "Learn Git in your browser for free with Try Git." To the right of the text is a diagram showing a branching model with stacks of code blocks connected by colored lines (blue, red, yellow). At the bottom, there are four sections: "About" (The advantages of Git compared to other source control systems.), "Documentation" (Command reference pages, Pro Git book content, videos and other material.), "Downloads" (GUI clients and binary releases for all major platforms.), and "Community" (Get involved! Bug reporting, mailing list, chat, development and more.). On the right side of the bottom section is a monitor displaying "Latest source Release 2.14.2", "Release Notes (2017-09-22)", and a button "Downloads for Linux".

Após o download,  
Next, Next, Next...

## Passo 2:

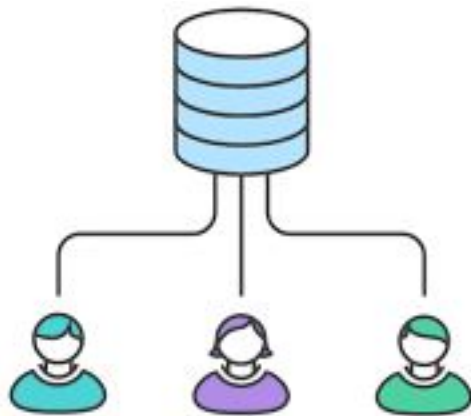
Criar sua conta no github:

<https://github.com>



Lembrar de aceitar o cadastro do github que foi enviado para o seu email.

# Clonando um repositório já existente no github



Podemos clonar qualquer repositório público ou privado, desde que nos seja permitido o acesso.




# Clonando um repositório já existente no github

1. No **terminal** (cmd, cmd, etc), navegue até a pasta onde deseja salvar o repositório;
2. Ainda no terminal, digite o seguinte comando:

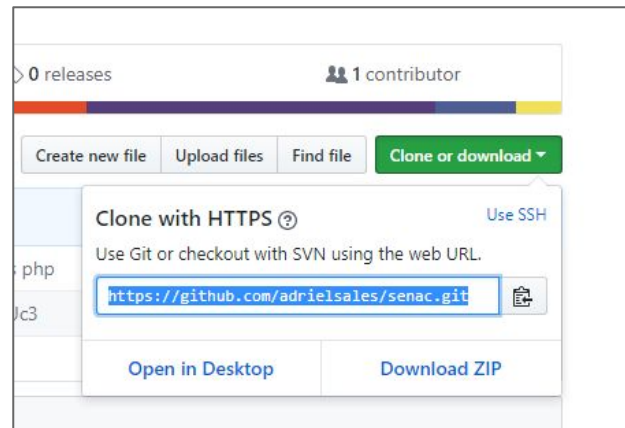
```
git clone https://github.com/adrielacademico/repo-name.git
```



Comando do git para clonar um repositório.



URL do repositório, copiado do github.



# Criando um repositório no github

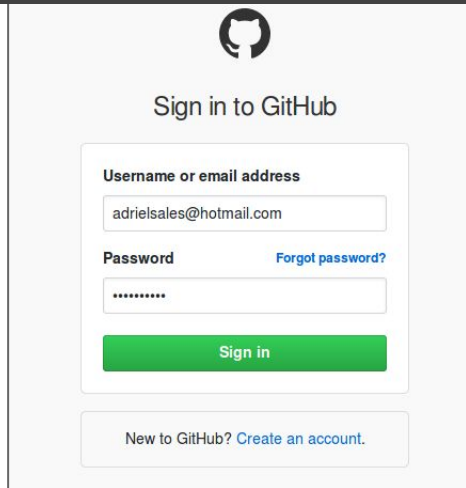


Acesse seu código em qualquer lugar!

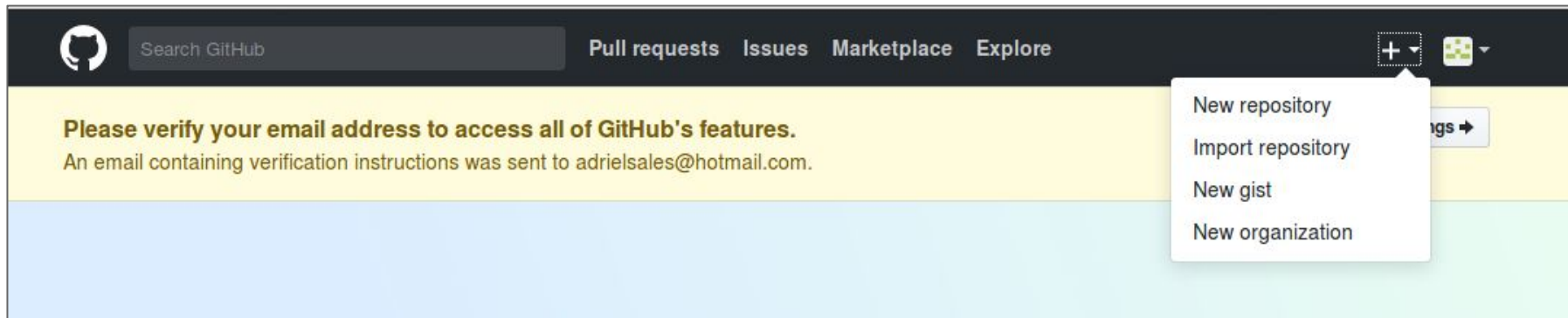


# Criando um repositório no github

1. Acesse sua conta no github.
2. Clique em + e new repository.



The image shows the GitHub sign-in interface. At the top is the GitHub logo and the text "Sign in to GitHub". Below this is a form with two input fields: "Username or email address" containing "adrielsales@hotmail.com" and "Password" with masked characters. A "Forgot password?" link is next to the password field. A green "Sign in" button is at the bottom of the form. Below the form is a link for "New to GitHub? Create an account."



# Criando um repositório no github

## Informe:

1. Nome do repositório;
2. A descrição;
3. Tipo (publico - **free \$**);
4. E clique em **criar!**


### Create a new repository

A repository contains all the files for your project, including the revision history.

---

Owner

Repository name


 **adrielaacademico** ▾ / **loja-programador-web** ✓

Great repository names are short and memorable. Need inspiration? How about **glowing-palm-tree**.


Description (optional)

Projeto de uma Loja Virtual para o Curso Programador Web do Senac PB (Unidade João Pessoa) 2017.

---

☒  **Public**

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

☐  **Private**


You choose who can see and commit to this repository.

---

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

---

**Create repository**

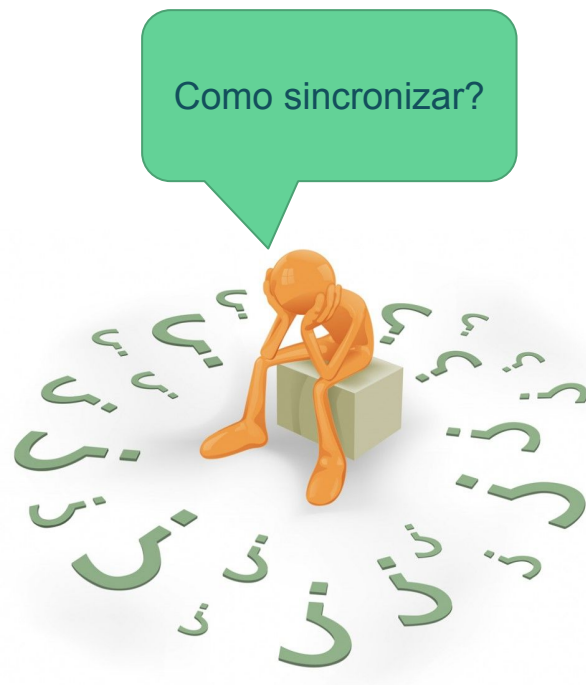
# Sincronizando um repositório

- **Repositório Local:**

- Está em nossa máquina: **git** 

- **Repositório Remoto:**

- Está na “nuvem”: **github** 



# Sincronizando um repositório

Lembre: para esse trabalho, usaremos o terminal

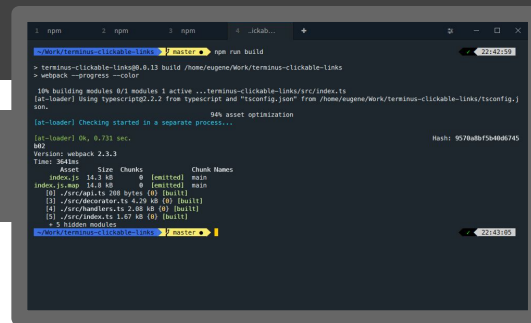
1. Crie um repositório no **github** e copie a sua URL;
2. **Em seu computador**, entre no diretório que deseja sincronizar.
3. Inicialize o git, informando que este diretório será um repositório git:

```
git init
```

4. Crie uma conexão entre o repositório local e o remoto, como comando:

```
git remote add origin
```

```
https://github.com/adrielsales/teste-sincronizar.git
```



```
1 npm 2 npm 3 npm 4 ...click... +
> terminus-clickable-links@0.1.3 build /home/eugene/work/terminus-clickable-links
> webpack --mode=production --color

10% building modules 8/1 modules 1 active ...terminus-clickable-links/src/index.ts
[webpack loader] using typescript@2.2.7 from typescript and "tsconfig.json" from /home/eugene/work/terminus-clickable-links/tsconfig.json.
94% asset optimization
[webpack loader] Checking started in a separate process...
[webpack loader] OK, 0.731 sec.
Hash: 907b4d7f5b4b4745
Version: webpack 2.3.3
Time: 364ms
Asset      Size      Chunks    [emitted]    Chunk Names
index.js  14.3 KB      0 [emitted]    main
index.js.map  38.8 KB      0 [emitted]    main
[0] ./src/api.ts 208 bytes [0] [built]
[1] ./src/decoder.ts 4.29 KB [0] [built]
[2] ./src/handlers.ts 2.68 KB [0] [built]
[3] ./src/index.ts 1.67 KB [0] [built]
+ 3 hidden modules
```

# Sincronizando um repositório

Use os comandos do git para:

- Baixe atualizações no repositório do github: `git pull origin master`
- Verifique o status do seu repositório local: `git status`
- Adicione arquivos ao controle de versão: `git add .`
- “Comite” os arquivos: `git commit -m “explica o que mudou”`
- Envie os arquivos para o github: `git push origin master`

# Comandos mais usados

## Localmente

- Iniciar um repositório git: `git init`
- Verificar o status do seu repositório: `git status`
- Adicionar arquivos ao controle de versão: `git add .`
- “Comitar” os arquivos: `git commit -m “mensagem”`



## Remotamente

- Baixar atualizações no repositório do github: `git pull origin master`
- Enviar arquivos para o github: `git push origin master`

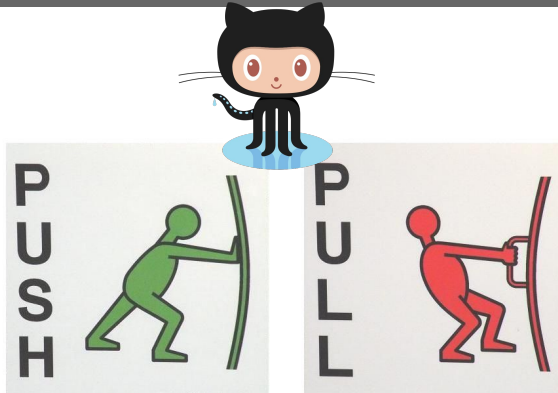


# Sincronizando repositórios

IMPORTANTE

**git push**

Empurra alterações do repositório local para o repositório remoto.



**git pull**

Trás as alterações que estão no repositório remoto para o repositório local.



# Configuração local do GIT

1. É necessário adicionar nome e email para gerenciar seu repositório local:
2. (use os mesmo do github):

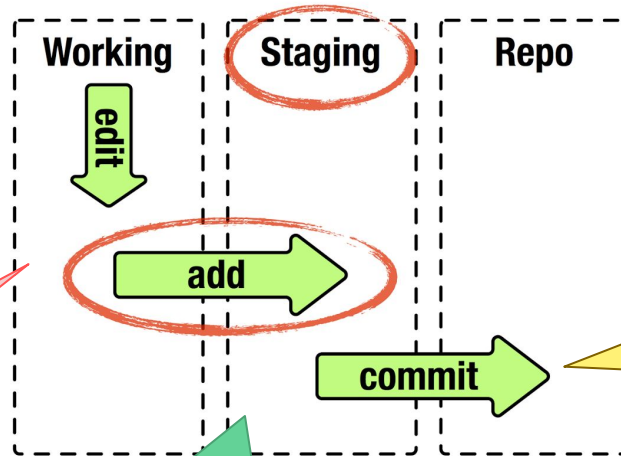
```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

3. Verificando as configurações:

```
$ git config --list  
user.name=Scott Chacon  
user.email=schacon@gmail.com  
color.status=auto  
color.branch=auto
```

Use **--global** apenas se pretende que todos os repositórios na máquina sejam gerenciados **pela mesma conta**, caso contrário, não precisa usar este parâmetro.

# Estágios do GIT



## WORKING

Os arquivos ainda não estão sendo monitorados pelo GIT.

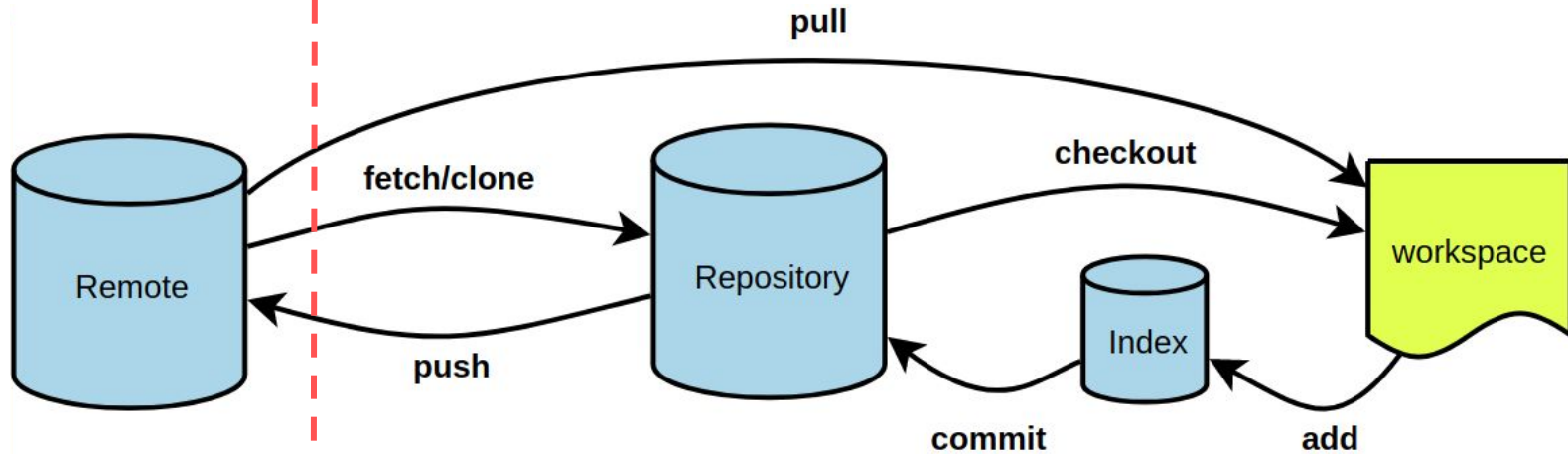
## STAGING

Após aplicar o comando **git add**, os arquivos passam a ser monitorados pelo git, mas ainda não estão prontos para serem salvos no repositório remoto.

## REPOSITORY

Após aplicar o comando **git commit -m "explicando as alterações"** os arquivos estão prontos para serem salvos no repositório remoto.

# Ciclo lógico do controle de versão usando git e github



Repositório  
Remoto - github

Repositório local - computador

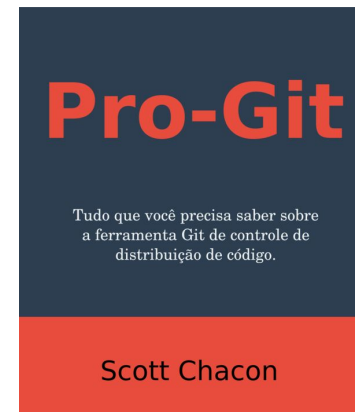
# Cursos e Livro Free!



<https://www.youtube.com/watch?v=WVLhm1AMeYE&list=PLInBAd9OZCzzHBJjLFZzRI6DgUmOeG3H0>



<https://www.udemy.com/git-e-github-para-iniciantes/>



<https://github.com/adrielsales/senac/tree/master/programadorWeb2017/free-books>



Professor:  
Adriel Sales



Fontes:



<https://git-scm.com>

<https://github.com>

