



# GitHub

## 101

# It's Me!



## Raysa Dutra

Software and Computer Engineer, Dog Lover and a Dreamer.

Organizadora da PythOnRio, Google Developers Group RJ, Diversity.js (JS Ladies RJ) e Diversity.linux (LinuxChix)

Embaixadora do Women Techmakers RJ

Co-Fundadora do Diversity.js (JS Ladies BR)

Criadora do [@EventoTI\\_RJ](#)

Backend Developer na Zoop

[@hi\\_hi\\_ray](#) | [hi-hi-ray.github.io](https://hi-hi-ray.github.io)

# GitHub



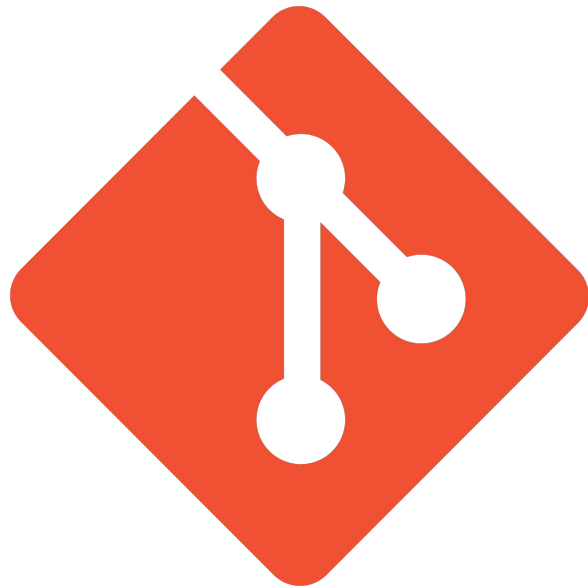
GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git.

Essa plataforma possui inúmeras funcionalidades, mas ela não é a única no mercado temos o BitBucket ou o GitLab, cada um oferecendo funcionalidades diferentes.

# Git

Git é um sistema de versionamento de arquivos criado pelo criador do Linux para ajudar ele a manter o código. Com ele você pode recuperar e atualizar os seus arquivos e etc...

O Git não é o único sistema de versionamento existentes ainda no mercado, nós temos em muitas empresas o SVN ou Mercurial, porém o Git é o mais famoso.

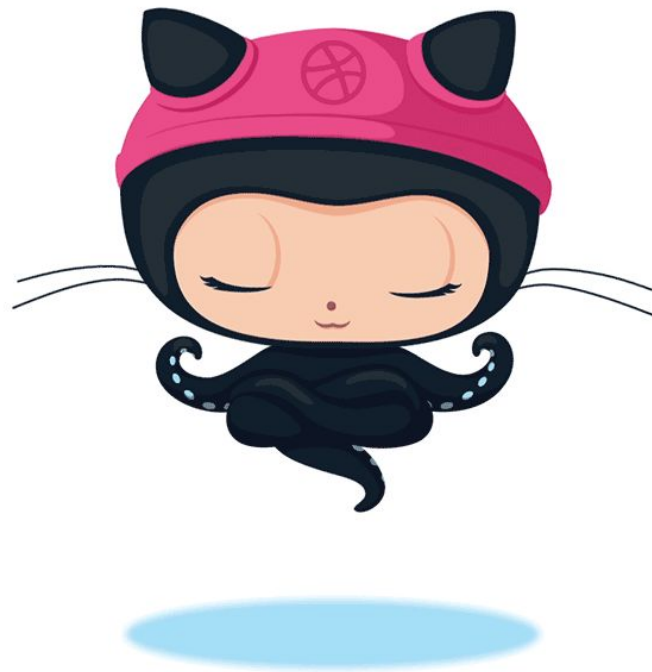


**Vamos  
por a mão  
na  
massa?**



# Check List

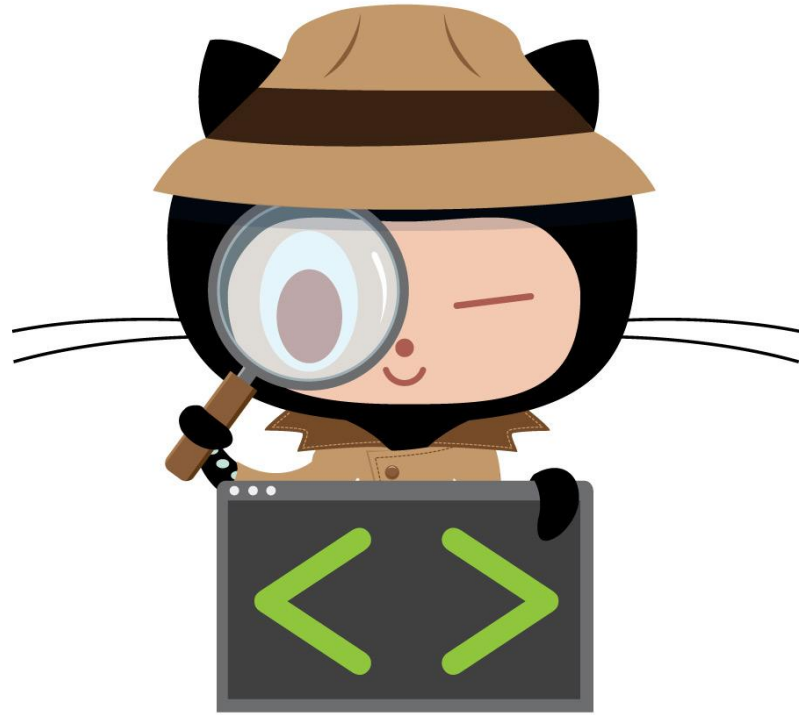
- Ter o Git Instalado.  
(<https://git-scm.com/>)
- Ter uma conta no Github.  
(<https://github.com>)





# Conhecendo o GitHub

# Vamos para o Terminal / Git Bash





# GIT CONFIG

O git config tem como função definir a sua Identidade, o que é muito importante para saber a identidade de quem está fazendo um commit.

**Primeiro comando:**

**git config --global user.name "Seu Nome"**

**Segundo comando:**

**git config --global user.email "Seu e-mail do git"**

# Comandos básicos



# GIT CLONE

O git clone tem como objetivo copiar um repositório existente para sua máquina, criando o ambiente de *workspace*.  
Existe 2 tipos de endereço o SSH ou o HTTPS.

**Base do comando:**

**git clone <endereço>**

# GIT STATUS

---

O git status tem como objetivo mostrar o estado atual do repositório que está no seu computador.

**Base do comando:**

**git status**

# GIT ADD

O git add tem como objetivo adicionar os arquivos criados/modificados no ambiente de *Stage (conhecido como index)*, ou seja, ele adiciona os arquivos em um local do *.git* do seu repositório.

**Base do comando:**

**git add <arquivo ou pasta ou .>**

# GIT COMMIT

O git commit tem como objetivo agrupar as mudanças que estão no *index* para o repositório local. Esses itens ficam na pasta *.git* criando um *blob* desses arquivos mudados.

Base do comando:

**git commit -m “mensagem”**

# GIT PUSH

---

O git push tem como objetivo mandar o seu *blob* para o repositório remoto.

Base do comando:

**git push**

# GIT PULL

---

O git pull atualiza o repositório que está no seu computador (*workspace*).

**Base do comando:**

**git pull <origin branch ou branch>**



# GIT FETCH

O git pull atualiza o seu repositório local (*local repository*). Esse comando pode ser utilizado toda vez que você desejar atualizar as branches do seu workspace.

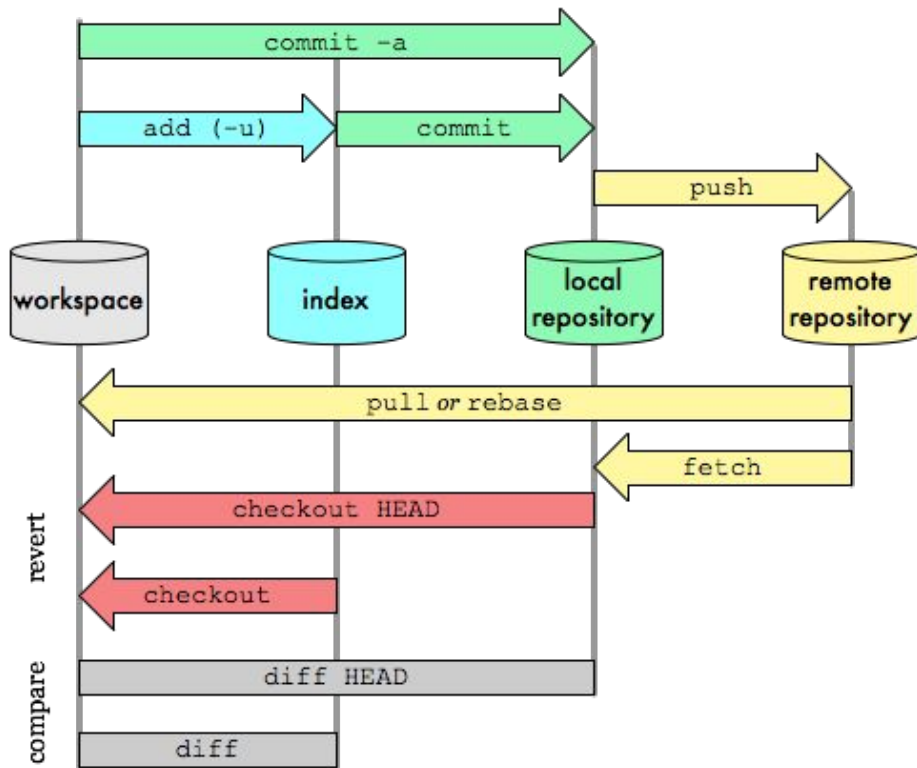
**Base do comando:**

**git fetch**



# O que aconteceu?

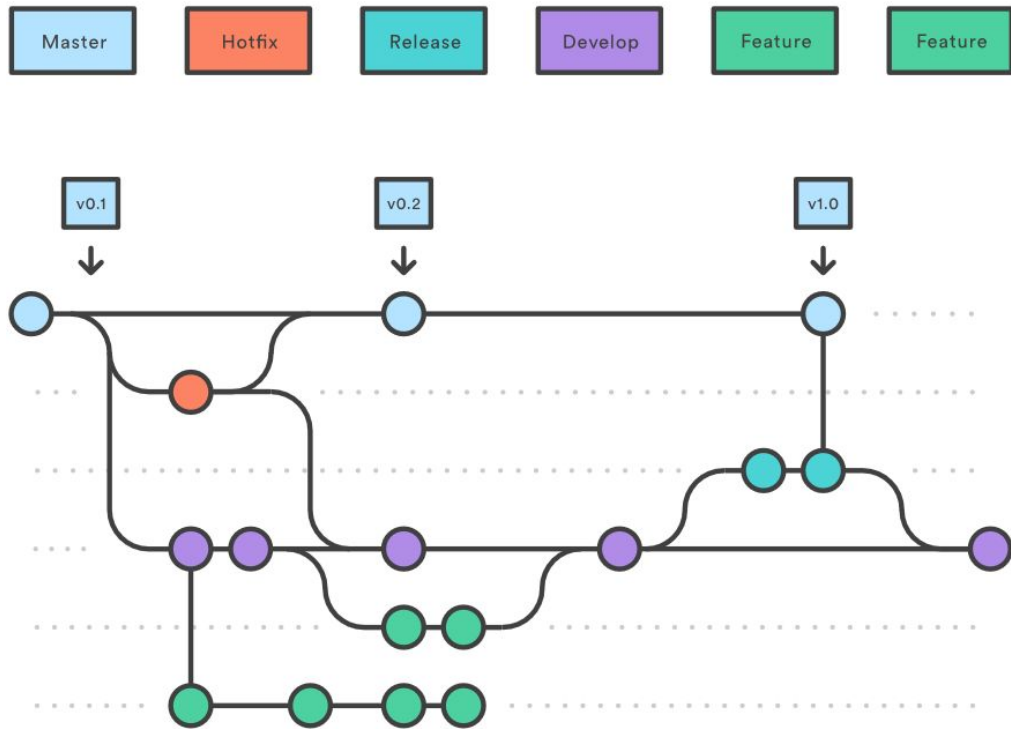
# COMANDOS NA TEORIA



# GIT WORKFLOW

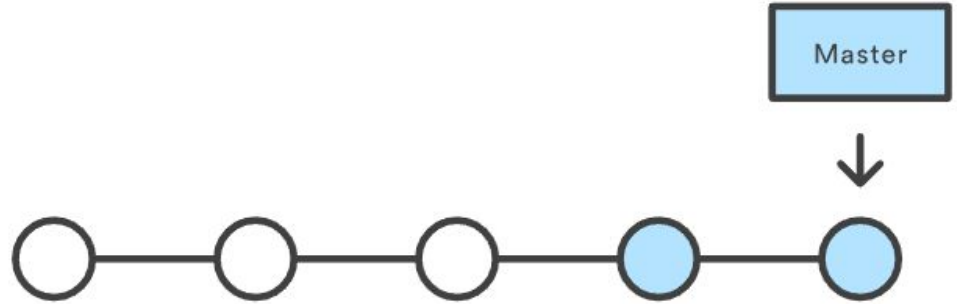
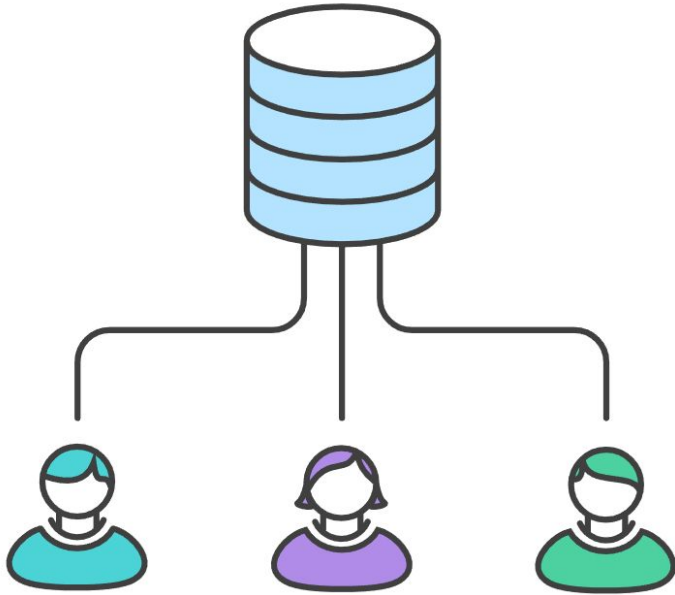


# GIT WORKFLOW

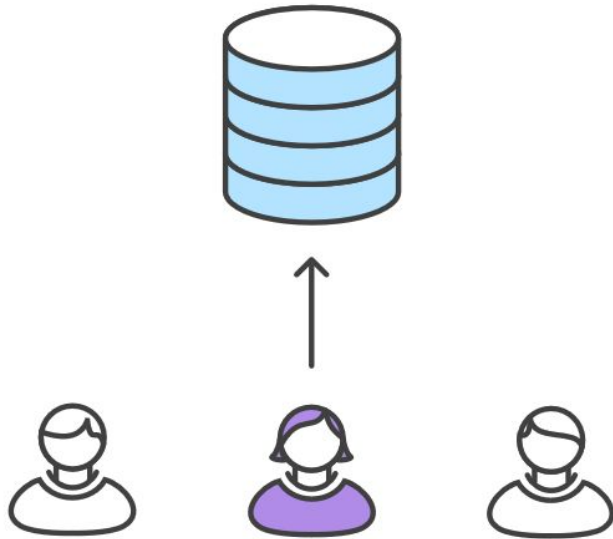


# WORKFLOW

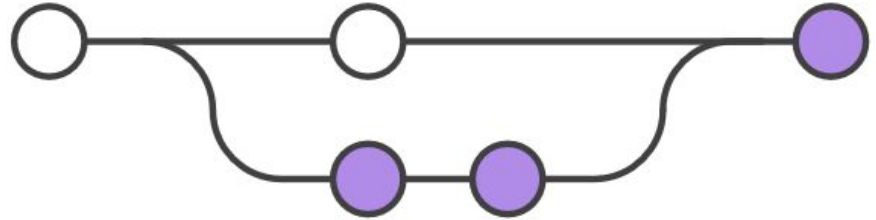
## Centralizado



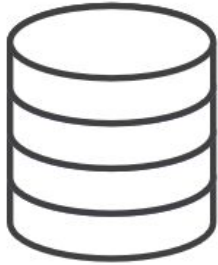
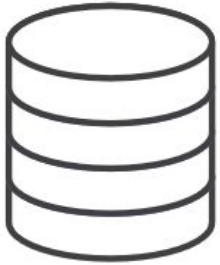
# WORKFLOW



## Feature



# WORKFLOW



**Fork**





# GIT BRANCH

As branches servem para nos ajudar a trabalhar de forma organizada sem quebrar o código que outra pessoa pode estar trabalhando.  
O comando abaixo te ensina a criar uma branch localmente.

**Base do comando:**

**git branch <nome da nova branch>**

# GIT CHECKOUT

Para ir para uma branch basta dar o comando abaixo.

Atenção! Se você tiver mudanças não adicionadas ele irá dar um erro, caso desejar descartar essas mudanças não salvas basta rodar o comando *git stash*.

**Base do comando:**

**git checkout <nome da branch>**

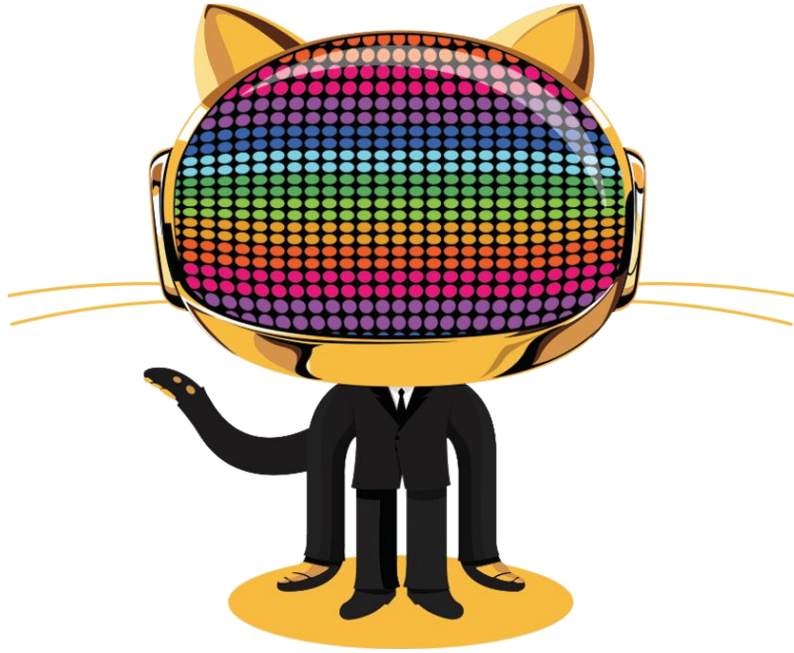
# GIT MERGE

Git merge tem como função juntar o que você fez em uma branch com outra branch.

Atenção! Você deve estar na branch que receberá as mudanças.

**Base do comando:**

**git merge <nome da branch que você quer juntar>**



# VAMOS PARA O GITHUB?

# GITHUB PARA ESTUDANTES

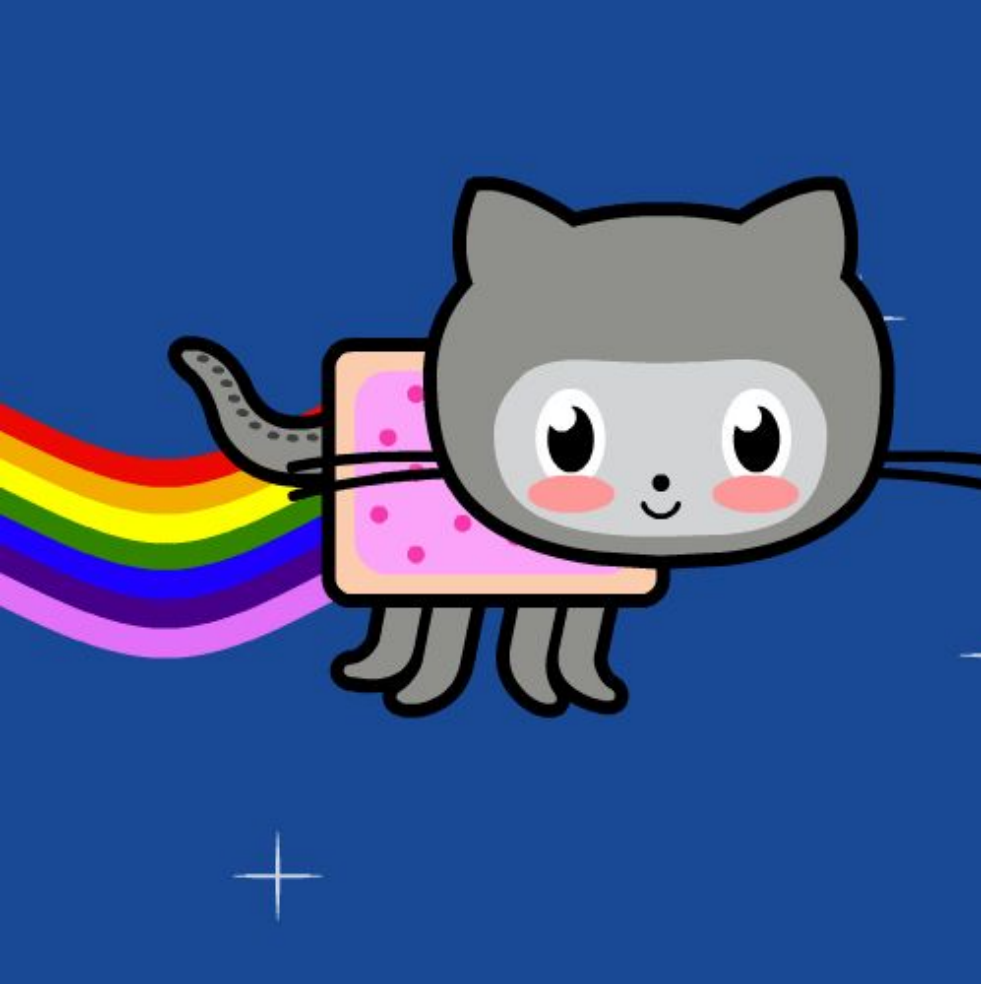


# EDUCATION PACK



**Cadastrem-se:**

<https://education.github.com/pack>



# MARKDOWN

# MARKDOWN



Assim como HTML é uma linguagem de marcação.

No GitHub ela é muito utilizado para documentar os códigos.

Segue alguns links úteis com exemplos de documentação, emoticons que vocês podem usar no commit e como formatar um markdown:

<https://is.gd/6h7FzY>



# OBRIGADA

