



# **Github para Gestão de Projetos e Empreendedores: Noções de Git.**

Gilvandro Medeiros, Cientista de Dados e Empreendedor

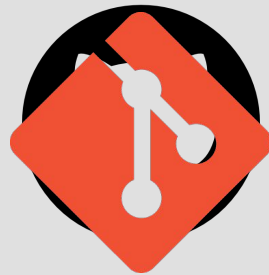
Contato: [gilvandro.cesar.mdrs@gmail.com](mailto:gilvandro.cesar.mdrs@gmail.com)

João Belau, Desenvolvedor e Empreendedor

Contato: [joao.santana@imobanco.com.br](mailto:joao.santana@imobanco.com.br) / [joaosb989@gmail.com](mailto:joaosb989@gmail.com)



# O que é Git?



O GitHub é uma excelente ferramenta, mas, para entendê-lo, precisamos saber de onde ele veio. Ele tem esse nome pois é filha da maior ferramenta de versionamento de código do mundo: o Git.

O Git foi desenvolvido por Linus Torvalds, em 2005, para suprir suas próprias necessidades de desenvolvimento de software. Linus utilizou o Git para versionar o seu maior projeto Open Source: o código fonte do Linux.



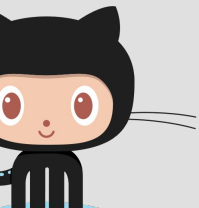
# O que é Git?



Mas, se o Git já é uma ferramenta por si só, onde entra o GitHub?

O Git é um software que não possui interface gráfica, apenas CLI (command line interface) e isso limita um pouco a visualização de suas funções, pois adiciona uma camada de muita abstração, além de precisar de um servidor local para servir a base dos repositórios.

Então, diversas iniciativas de criar uma interface gráfica e disponibilizar o repositório online para o Git começaram a aparecer. Hoje, a mais utilizada delas é o GitHub, embora não seja a única!



# Git o que?



GitHub

Gitlab



Gitea



Gogs



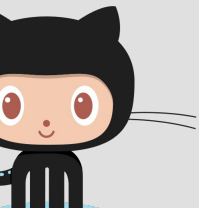
Phabricator



Redmine

# Dicionário Git

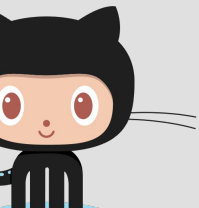
- Repositório: a pasta / endereço do projeto git.
- Commit: conjunto de alterações no projeto.
- Branch: ramificação no projeto.
- Merge: união de duas branches.
- Clone: uma cópia local do repositório principal.
- Pull: uma atualização do repositório local trazidas do principal.
- Push: enviar modificações do repositório local para o principal.
- Fork: é um clone do repositório para outro repositório online, onde ele terá suas próprias branches e modificações e pode seguir para um caminho de modificações totalmente diferente do original.
- Pull Request: pedido que se faz ao repositório para que uma branch seja adicionada ao projeto principal



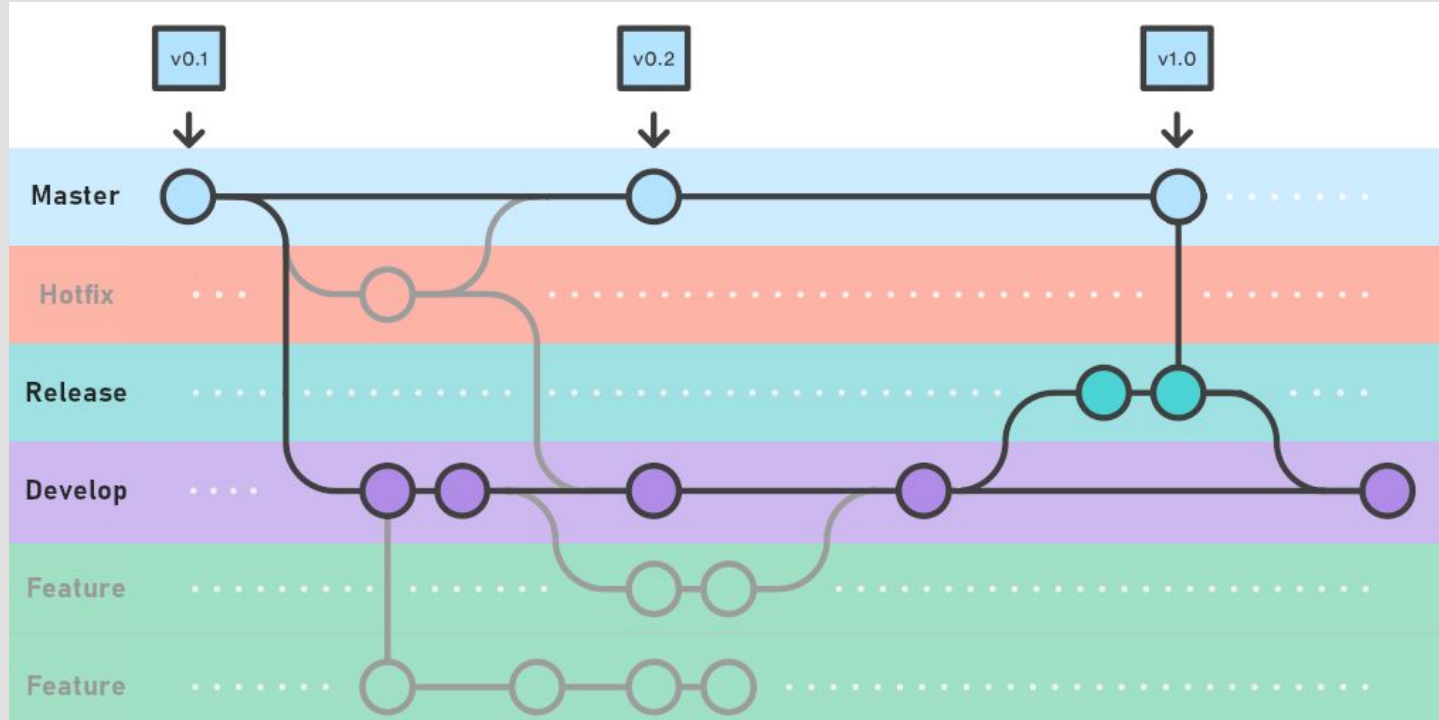
# Git

- O Git tem uma base que se chama REPOSITÓRIO
- Os repositórios têm branches (ramos)
- Cada ramo representa uma alteração no projeto
- Cada ramo pode ser versionado separadamente
- No final, os ramos devem ser comparados para serem mergidos (mesclados)
- Corrigem-se os conflitos e mesclam-se todos para o ramo principal

Ao final de tudo, é possível voltar para o ponto de cada modificação do projeto!

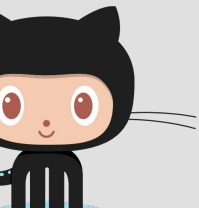


# Git Flow



# Git CLI

Se você não for familiarizado com linhas de comando, o GitHub pode facilitar, e muito, a sua vida.



```
joao@joao-Ryzen7:~/projeto$ git init -b main
Initialized empty Git repository in /home/joao/projeto/.git/
joao@joao-Ryzen7:~/projeto$ git config --global init.defaultBranch main
joao@joao-Ryzen7:~/projeto$ echo Olá, git >> arquivo1.txt
joao@joao-Ryzen7:~/projeto$ git add .
joao@joao-Ryzen7:~/projeto$ git commit -m "Adiciona primeiro arquivo ao repositório!"
[main (root-commit) bd12d45] Adiciona primeiro arquivo ao repositório!
 1 file changed, 1 insertion(+)
 create mode 100644 arquivo1.txt
joao@joao-Ryzen7:~/projeto$ git checkout -b feature/ola_mundo
Switched to a new branch 'feature/ola_mundo'
joao@joao-Ryzen7:~/projeto$ echo Olá, mundo! >> arquivo2.txt
joao@joao-Ryzen7:~/projeto$ git add .
joao@joao-Ryzen7:~/projeto$ git commit -m "Adiciona olá mundo com novo arquivo!!"
git commit -m "Adiciona olá mundo com novo arquivo"
[feature/ola_mundo 494ee5e] Adiciona olá mundo com novo arquivo
 1 file changed, 1 insertion(+)
 create mode 100644 arquivo2.txt
joao@joao-Ryzen7:~/projeto$ git checkout main
Switched to branch 'main'
joao@joao-Ryzen7:~/projeto$ git merge feature/ola_mundo
Updating bd12d45..494ee5e
Fast-forward
 arquivo2.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 arquivo2.txt
joao@joao-Ryzen7:~/projeto$ git log
commit 494ee5ecb5538f24af57a1016fbcf0b086a0b1af (HEAD -> main, feature/ola_mundo)
Author: João Santana <joaosb989@gmail.com>
Date: Tue Jan 18 23:47:24 2022 -0300

    Adiciona olá mundo com novo arquivo

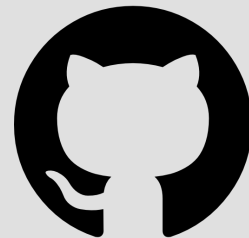
git add .

commit bd12d45d4bf684c077a7b8034be4808d49968a20
Author: João Santana <joaosb989@gmail.com>
Date: Tue Jan 18 23:46:37 2022 -0300

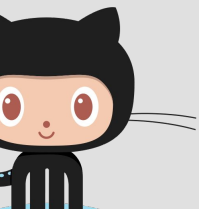
    Adiciona primeiro arquivo ao repositório!
joao@joao-Ryzen7:~/projeto$
```



# E não...



Novamente, não. Esse não é um tutorial (apenas) para programadores.



# Conceitos Aplicados



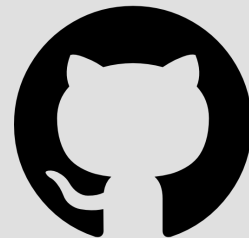
Pasta sem versionamento



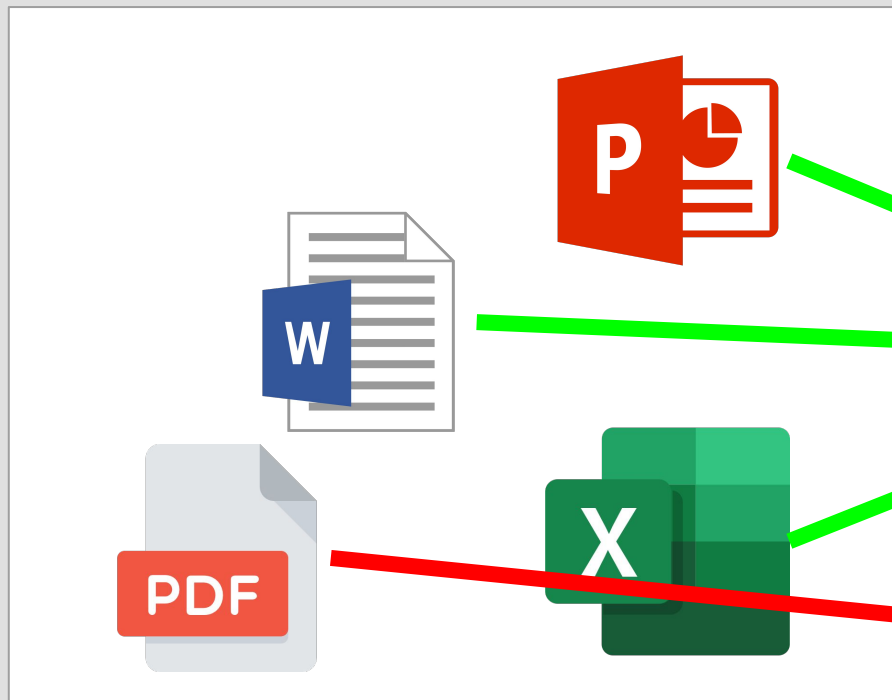
Passo 1: Iniciando git

**git init**

# Conceitos Aplicados



Pasta sem versionamento



Passo 2:

Selecionando elementos para serem versionados

**git add**

**git ignore**

# Conceitos Aplicados



Elementos sendo versionados



Passo 3:

“Commitando” projeto - armazenando versão atual dos arquivos monitorados pelo git.

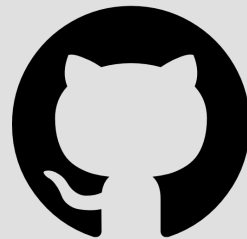
```
git commit -m “atualizacao  
do projeto”
```

# Conceitos Aplicados

Elementos sendo versionados



**git push**

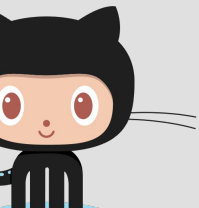


**Passo 4:**  
Enviando commits do repositório local para o repositório global - Github.

# Projetos

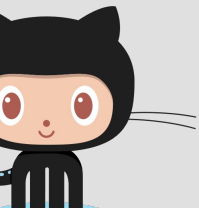
Apesar de ter sido um software focado no desenvolvimento de outros softwares, hoje podemos utilizar as ferramentas baseadas em Git para o desenvolvimento de projetos que não necessariamente envolvem códigos.

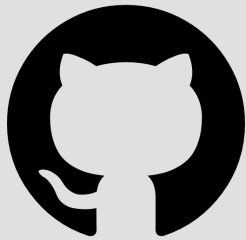
Além da ferramenta de versionamento, o GitHub possui ferramentas para gestão de projetos, onde pode-se aplicar Kanban, Scrum e outras metodologias ágeis.



# Projetos

No próximo vídeo, ensinaremos como iniciar um projeto pelo GitHub, trabalhar com arquivos que não sejam necessariamente código e aplicar técnicas de gerência através das ferramentas disponibilizadas lá.





```
~$ git add motivacao.cpp  
~$ git ignore preguica.dll  
~$ git commit -m "Adiciona motivacao para entender o Github"  
~$ git push
```

