

Git

@sandrovalerius



GIT, O QUE É?



O QUE É GIT?

- Atualmente é o sistema de controle de versão **mais utilizado do mundo**;
- O Git é baseado em **repositórios**, que contêm todas as versões do código e também as cópias de cada desenvolvedor;
- Todas as operações do Git **são otimizadas para ter alto desempenho**;
- Todos os objetos do Git são **protegidos por criptografia**;
- O Git é um **projeto de código aberto**;
- O Git **não é o GitHub**;



O QUE É CONTROLE DE VERSÃO?

- Uma técnica que ajuda a **gerenciar o código-fonte** de uma aplicação;
- Registrando **todas as modificações** de código, podendo também reverter as mesmas;
- Criar versões de um software em diferentes estágios, podendo **alterar facilmente entre elas**;
- Cada membro da equipe pode trabalhar em uma versão diferente;
- Há ferramentas para trabalhar o controle de versão como: **git** e SVN



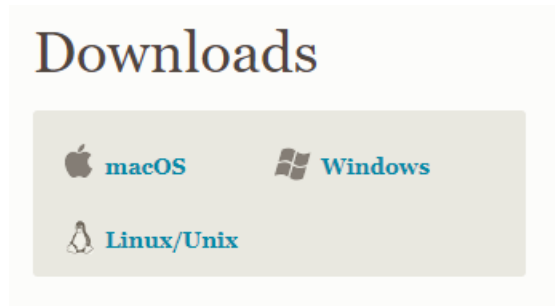
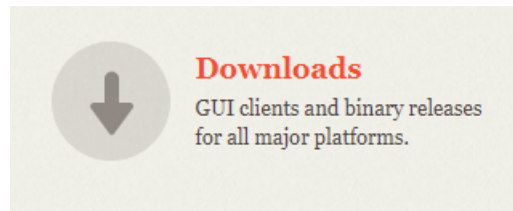


INSTALAÇÃO

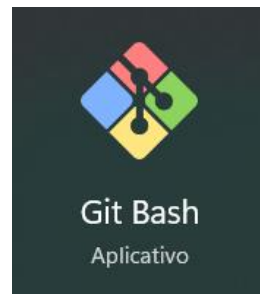


INSTALAÇÃO

- Acesse <https://git-scm.com>
- Downloads
- Escolher o Sistema Operacional



- Instalado o Git e o Terminal



GIT NO CONTEXTO DO SO

@sandrovalerius



```
C:\Users\sandr\AppData\Roam X + v

D:\www>path
PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\dotnet\;C:\Program Files\Git\cmd;C:\Users\sandr\AppData\Local\Programs\Python\Python39\Scripts\;C:\Users\sandr\AppData\Local\Programs\Python\Python39\;C:\Users\sandr\AppData\Local\Microsoft\WindowsApps;;C:\Users\sandr\AppData\Local\Programs\Microsoft VS Code\bin;C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.1\bin;;C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin;

D:\www>|
```

GIT NO CONTEXTO DO SO

```
C:\Users\sandr\AppData\Roam... X + v
D:\www>path
PATH=C:\WINDOWS\system32;C:\WINDOWS\system32\Wbem;C:\WINDOWS\system32\WindowsPowerShell\v1.0\;C:\Program Files\dotnet\;C:\Program Files\Git\cmd;C:\Users\sandr\AppData\Local\Programs\Python\Python39\;C:\Users\sandr\AppData\Local\Programs\Microsoft VS Code\bin;C:\Program Files\JetBrains\PyCharm\bin;C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3.2\bin;
```



GIT BASH

@sandrovalerius



```
MINGW64:/c/Users/sandr
Sandro Valérius@Sandro MINGW64 ~ (main)
$ |
```

GIT

cmd

```
D:\www>git --version  
git version 2.36.1.windows.1  
  
D:\www>|
```

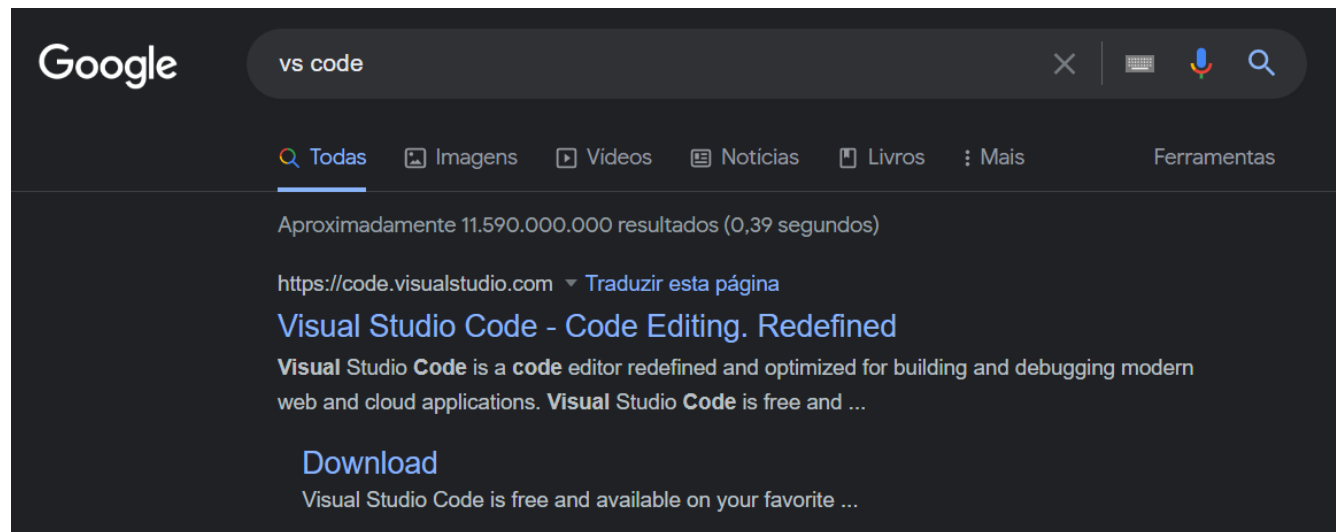
Git Bash

```
MINGW64:/c/Users/sandr  
  
Sandro Valérius@Sandro MINGW64 ~ (main)  
$ git --version  
git version 2.36.1.windows.1  
  
Sandro Valérius@Sandro MINGW64 ~ (main)  
$ |
```



INSTALAÇÃO DO *VS CODE*

- O VS Code possui um terminal integrado, facilitando as ações com o **git**;
- Além de ser um editor super atualizado e que aceita diversas linguagens e ferramentas de programação;



INSTALAÇÃO DO *VS CODE*

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 8, 10, 11

User Installer	64 bit	32 bit	ARM
System Installer	64 bit	32 bit	ARM
.zip	64 bit	32 bit	ARM



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

.deb	64 bit	ARM	ARM 64
.rpm	64 bit	ARM	ARM 64
.tar.gz	64 bit	ARM	ARM 64
Snap Store			



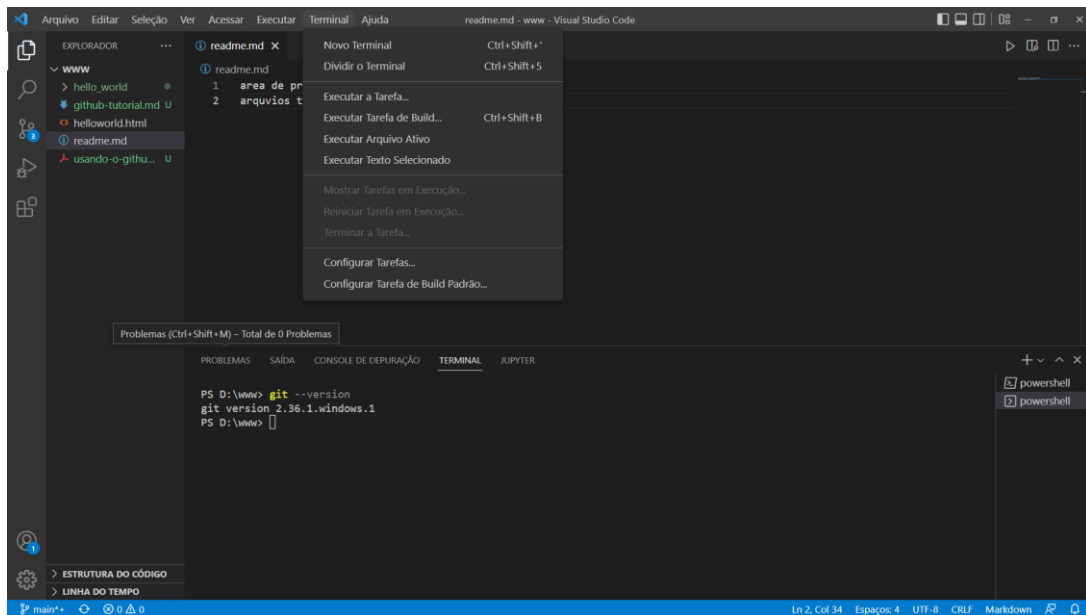
↓ Mac

macOS 10.11+

.zip [Universal](#) [Intel Chip](#) [Apple Silicon](#)

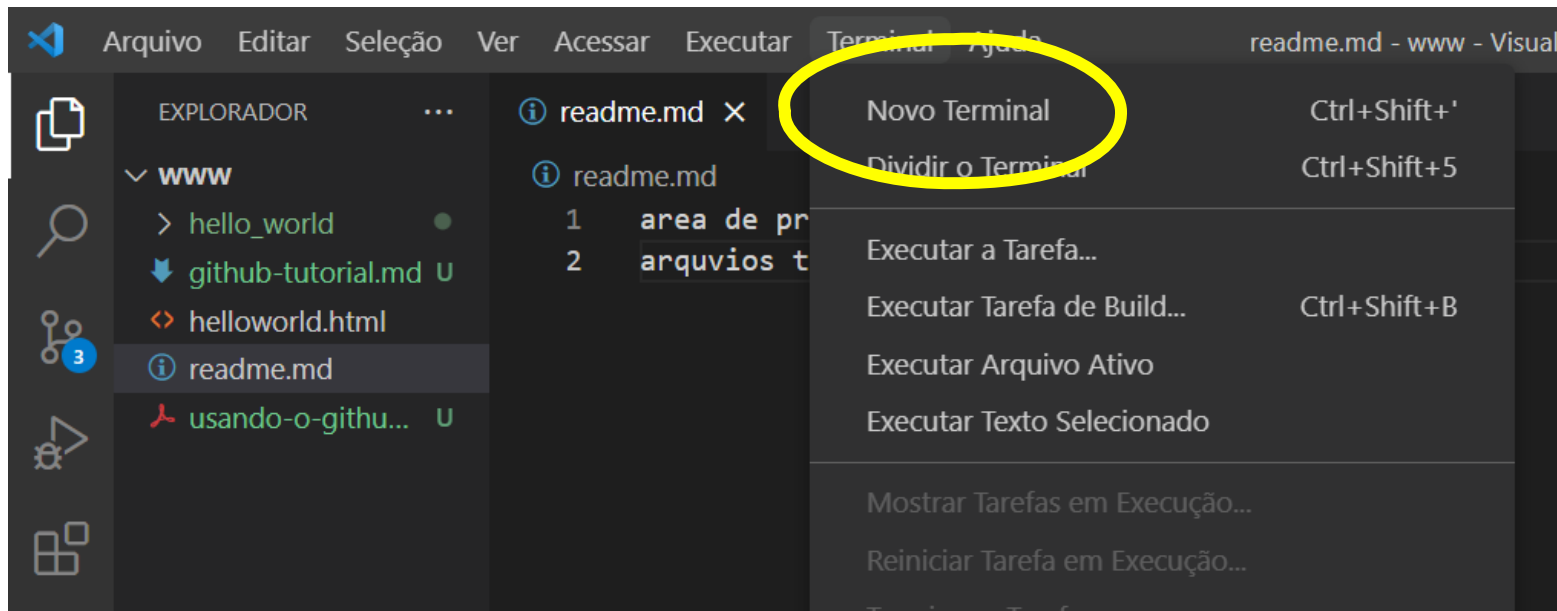


TERMINAL DO VS CODE



TERMINAL DO VS CODE

@sandrovalerius



TERMINAL DO *VS CODE*

PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

JUPYTER

```
PS D:\www> git --version  
git version 2.36.1.windows.1  
PS D:\www> 
```





REPOSITÓRIO



O QUE É UM REPOSITÓRIO?

- Local onde o código será **armazenado**;
- Cada projeto tem **um repositório**;
- Ao criar um repositório, iniciando um projeto;
- O repositório é carregado em servidores que gerenciam repos, como: **GitHub** e **Bitbucket**;
- Os desenvolvedores do time podem baixar o repositório e **criar versões diferentes** localmente;



CRIANDO REPOSITÓRIOS

- Para criar um repositório utilizamos o comando: **git init**
- Os arquivos do Git estão numa pasta oculta **.git**;
- Após este comando o diretório atual **será reconhecido pelo git como um projeto** e responderá aos seus demais comandos;

```
D:\www\projetoTeste>git init  
Initialized empty Git repository in D:/www/projetoTeste/.git/
```



CRIANDO REPOSITÓRIOS

- Para criar um repositório utilizamos o comando: **git init**
- Os arquivos do Git estão numa pasta oculta **.git**;
- Após este comando o diretório atual **será reconhecido pelo git como um projeto** e responderá aos seus demais comandos;



```
D:\www\projetoTeste>dir /a
O volume na unidade D é Dados
O Número de Série do Volume é 94DD-7346

Pasta de D:\www\projetoTeste

26/08/2022  15:59    <DIR>          .
26/08/2022  15:57    <DIR>          ..
26/08/2022  15:59    <DIR>          .git
                0 arquivo(s)                0 bytes
                3 pasta(s)      565.632.409.600 bytes disponíveis
```



GITHUB

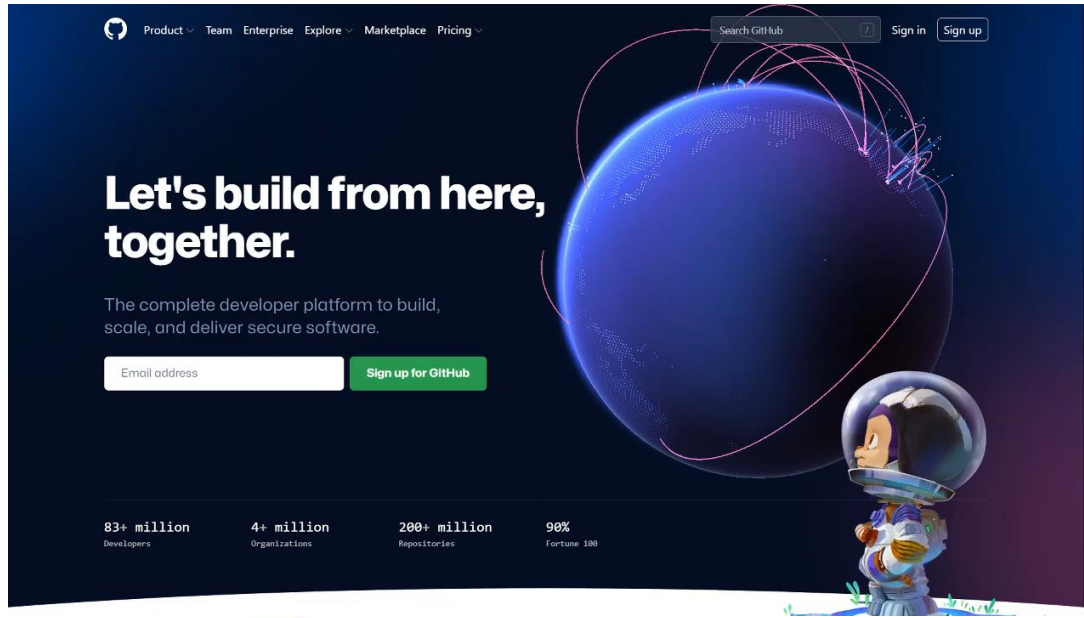


O QUE É O GITHUB?

- É um **serviço gratuito** para gerenciar repositórios;
- Podemos **enviar nossos projetos** para o GitHub e disponibilizá-lo para outros devs;
- O GitHub é **gratuito** tanto para projetos públicos como **privados**;
- <https://github.com>



O QUE É O GITHUB?



- <https://github.com>

É UM PROCESSO
BEM COMUM
DE CADASTRO

CONFIRME A CONTA
PELO E-MAIL



ENVIANDO REPOSITÓRIOS



- **Criar** o repositório no GitHub;
- **Inicializar** o mesmo repositório no Git do computador;
- **Sincronizar** com o GitHub;
- **Enviar** o projeto;

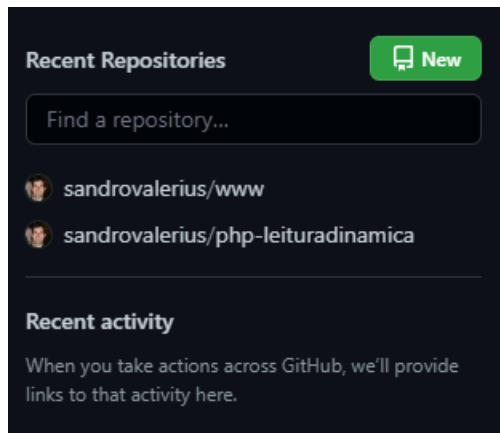


CALMA!
NÃO SÃO TANTOS
COMANDOS ASSIM!

ATENÇÃO
ESTE FLUXO É SÓ NA
CRIAÇÃO DO PROJETO!

ENVIANDO REPOSITÓRIOS

- **Criar** o repositório no GitHub;
- **Inicializar** o mesmo repositório no Git do computador;
- **Sincronizar** com o GitHub;
- **Enviar** o projeto;



ENVIANDO REPOSITÓRIOS


- **Criar** o repositório no GitHub;
- **Inicializar** o mesmo repositório no Git do computador;
- **Sincronizar** com o GitHub;
- **Enviar** o projeto;



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 *


 sandrovalerius / projetoTeste ✓

Great repository names are short and memorable. Need inspiration? How about [urban-tribble?](#)

Description (optional)

Repositório para o teste na Fated

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.


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

Create repository

ENVIANDO REPOSITÓRIOS

@sandrovalerius

Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

<https://github.com/sandrovalerius/projetoTeste.git>



Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# projetoTeste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code



ENVIANDO REPOSITÓRIOS

- **Criar** o repositório no GitHub;
- **Inicializar** o mesmo repositório no Git do computador;
- **Sincronizar** com o GitHub;
- **Enviar** o projeto;



```
D:\www\projetoTeste>git init
Initialized empty Git repository in D:/www/projetoTeste/.git/

D:\www\projetoTeste>dir /a
O volume na unidade D é Dados
O Número de Série do Volume é 94DD-7346

Pasta de D:\www\projetoTeste

26/08/2022  15:59    <DIR>          .
26/08/2022  15:57    <DIR>          ..
26/08/2022  15:59    <DIR>          .git
                0 arquivo(s)                0 bytes
                3 pasta(s)  565.632.409.600 bytes disponíveis
```

ENVIANDO REPOSITÓRIOS

- **Criar** o repositório no GitHub;
- **Inicializar** o mesmo repositório no Git do computador;
- **Sincronizar** com o GitHub;
- **Enviar** o projeto;



```
D:\www\projetoTeste>git init
Initialized empty Git repository in D:/www/projetoTeste/.git/
```

```
D:\www\projetoTeste>dir /a
O volume na unidade D é Dados
O Número de Série do Volume é 94DD-7346
```


```
Pasta de D:\www\projetoTeste
```

```
26/08/2022  15:59    <DIR>          .
26/08/2022  15:57    <DIR>          ..
26/08/2022  15:59    <DIR>          .git
                0 arquivo(s)                0 bytes
                3 pasta(s)  565.632.409.600 bytes disponíveis
```

ENVIANDO REPOSITÓRIOS

@sandrovalerius

Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

<https://github.com/sandrovalerius/projetoTeste.git>



Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# projetoTeste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git branch -M main
git push -u origin main
```



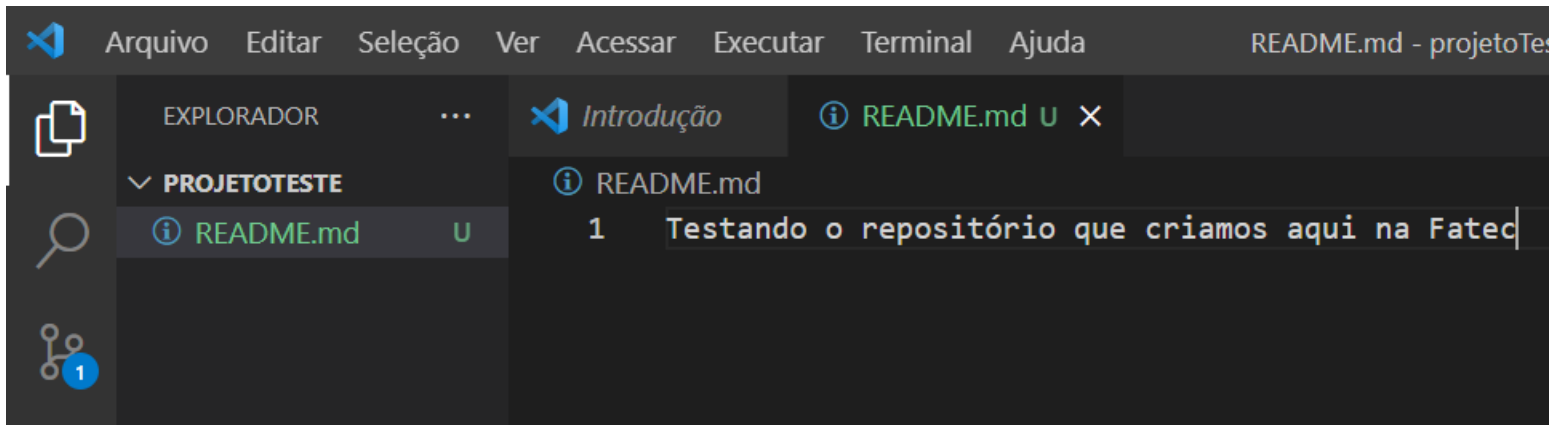
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code



ENVIANDO REPOSITÓRIOS



```
echo "# projetoTeste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git push -u origin main
```

ENVIANDO REPOSITÓRIOS

PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

JUPYTER

```
PS D:\www\projetoTeste> git status
```

```
On branch master
```

```
No commits yet
```


```
Untracked files:
```

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

```
README.md
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
PS D:\www\projetoTeste> █
```



```
echo "# projetoTeste" >> README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
```

```
git push -u origin main
```

ENVIANDO REPOSITÓRIOS

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL JUPYTER

```
PS D:\www\projetoTeste> git add README.md
PS D:\www\projetoTeste> git status
On branch master

No commits yet

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

PS D:\www\projetoTeste> █
```



```
echo "# projetoTeste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git push -u origin main
```


ENVIANDO REPOSITÓRIOS

PROBLEMAS


SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

JUPYTER

```
PS D:\www\projetoTeste> git commit -m "primeiro projeto"
[master (root-commit) af3a04f] primeiro projeto
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
PS D:\www\projetoTeste> git status
On branch master
nothing to commit, working tree clean
PS D:\www\projetoTeste> █
```



```
echo "# projetoTeste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git push -u origin main
```

ENVIANDO REPOSITÓRIOS

PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

JUPYTER

```
PS D:\www\projetoTeste> git branch -M main
```

```
PS D:\www\projetoTeste> 
```

```
echo "# projetoTeste" >> README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
```

```
git push -u origin main
```

ENVIANDO REPOSITÓRIOS

PROBLEMAS


SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

JUPYTER

```
PS D:\www\projetoTeste> git remote add origin https://github.com/sandrovalerius/projetoTeste.git
PS D:\www\projetoTeste> █
```



```
echo "# projetoTeste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git push -u origin main
```

ENVIANDO REPOSITÓRIOS

- **Criar** o repositório no GitHub;
- **Inicializar** o mesmo repositório no Git do computador;
- **Sincronizar** com o GitHub;
- **Enviar** o projeto;



ENVIANDO REPOSITÓRIOS

PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL


JUPYTER


```
PS D:\www\projetoTeste> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 270 bytes | 270.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sandrovalerius/projetoTeste.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS D:\www\projetoTeste> █
```

```
echo "# projetoTeste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sandrovalerius/projetoTeste.git
git push -u origin main
```

ENVIANDO REPOSITÓRIOS

@sandrovalerius

 main ▾


 1 branch

 0 tags

[Go to file](#)

[Add file ▾](#)

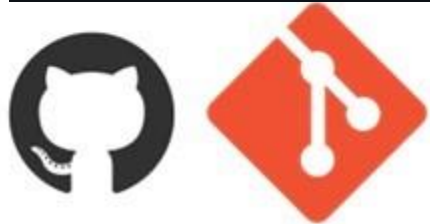
[Code ▾](#)

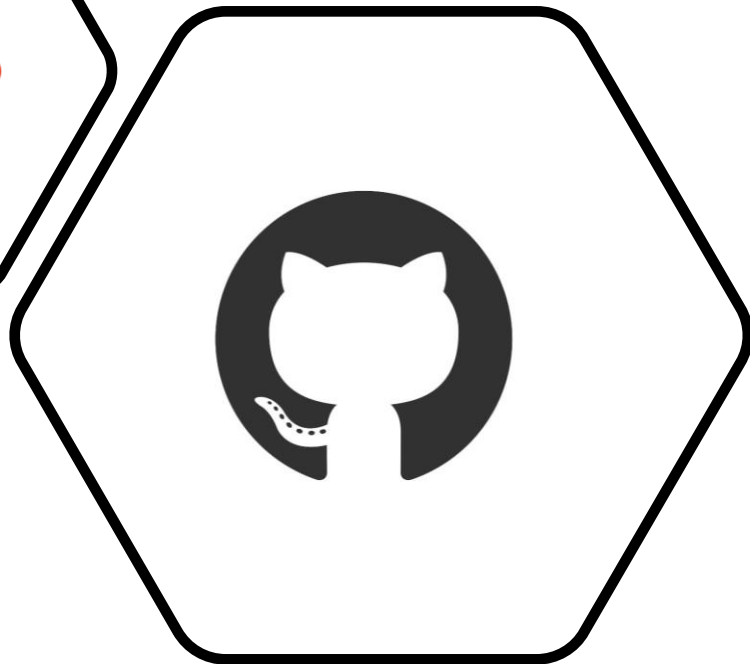
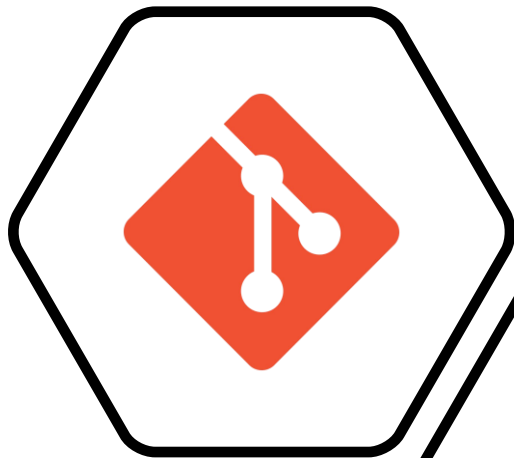
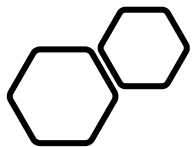
 **sandrovalerius** primeiro projeto af3a04f 12 minutes ago  1 commit

 README.md primeiro projeto 12 minutes ago

README.md 

Testando o repositório que criamos aqui na Fatec





Git

@sandrovalerius