

GIT E
GITHUB

O que é o Git?



O Git é um software de versionamento de CÓDIGO que funciona de forma distribuída.

O Git foi criado por Linus Torvalds criador do kernel Linux. O Git foi criado para auxiliar no desenvolvimento do kernel Linux e também uma alternativa ao BitKeeper por ser um software proprietário.

Existe outros software de versionamento como o CVS, Subversion que diferente do Git funcionam de forma concorrente.

O que é o GitHub?



O GitHub é a maior plataforma de hospedagem de código-fonte do mundo, o funcionamento dele é de um repositório remoto para Git. O GitHub surgiu em 2008, o mesmo foi desenvolvido por Chris Wanstrath, J. Hyett, Tom Preston-Werner e Scott Chacon.

Existem algumas alternativas ao GitHub como o GitLab, BitBucket, Gogs, Gitea entre outros.

Primeiro Repositorio GIT

No terminal do Linux coloque o seguinte comando para criar uma pasta vazia:

```
[italo@7K tutoriais]$ mkdir tutorial_git
```

Entre na pasta e execute o comando para iniciar um repositorio Git:

```
[italo@7K tutoriais]$ cd tutorial_git  
[italo@7K tutorial_git]$ git init
```


Depois de dar o comando git init aparecera a seguinte mensage:

```
Initialized empty Git repository in /run/media/italo/DATA/Developm  
ent/tutoriais/tutorial_git/.git/
```

O comando git init é usado para criar um repositorio git, ele cria uma pasta chamda .git, onde está todo o historico do seu programa.

Repositorio Do GitHub

Para fazer o link entre o repositorio local e o novo repositorio basta criar um repositorio no GitHub e copiar o link do repositorio.

Owner:  italonicacio / Repository name *

Great repository names are short and memorable. Need inspiration? How about [furry-octo-bassoon](#)?

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

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

or **HTTPS** **SSH** `https://github.com/italonicacio/Tutorial-Git.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 "# Tutorial-Git" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/italonicacio/Tutorial-Git.git
git push -u origin master
```

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

```
git remote add origin https://github.com/italonicacio/Tutorial-Git.git
git push -u origin master
```

...or import code from another repository

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

Import code

Repositorio Do GitHub

Para adicionar o link do repositorio remoto do GitHub no git basta fazer o comando abaixo:

```
[italo@7K tutorial_git]$ git remote add origin https://github.com/italonicacio/Tutorial-Git.git
```

Esse comando adiciona um repositorio remoto.

Agora vamos criar um arquivo e adicionar ele nos arquivos em que o Git vai monitorar para ver se tem modificações.

```
[italo@7K tutorial_git]$ >tutorial.py
```

Apos isso coloque alguma coisa no arquivo, nesse exemplo usarei um comando básico para ilustrar.

Repositorio Do GitHub

Para adicionar o link do repositorio remoto do GitHub no git basta fazer o comando abaixo:

```
tutorial.py ×  
tutorial.py  
1 print("Hello Git")  
2 |
```

Apos criar o arquivo se você executar o git status no terminal ira aparecer isso:

```
[italo@7K tutorial_git]$ git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    tutorial.py  
  
nothing added to commit but untracked files present (use "git add" to track)
```

Repositorio Do GitHub

Para adicionar o link do repositorio remoto do GitHub no git basta fazer o comando abaixo:

```
tutorial.py ×
tutorial.py
1 print("Hello Git")
2 |
```

Apos criar o arquivo se você executar o git status no terminal ira aparecer isso:

```
[italo@7K tutorial_git]$ git status
On branch master

No commits yet

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

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


Repositorio Do GitHub

Agora basta fazer o comando `git add` para adicionar o novo arquivo ou uma nova pasta com arquivos, para o repositorio git.

```
[italo@7K tutorial_git]$ git add tutorial.py
```

Um commit é dado quando você quer marcar um ponto da historia, ou seja, quando você acha que esta em uma parte importante que precisa ser salvo para se der problema posteriormente, voltar para esse ponto no tempo.

```
[italo@7K tutorial_git]$ git commit -m "primeiro commit"
[master (root-commit) af07e19] primeiro commit
1 file changed, 1 insertion(+)
create mode 100644 tutorial.py
```

Repositorio Do GitHub

Se você der outro `git status` aparecerá a seguinte mensagem:

```
[italo@7K tutorial_git]$ git status  
On branch master  
nothing to commit, working tree clean
```

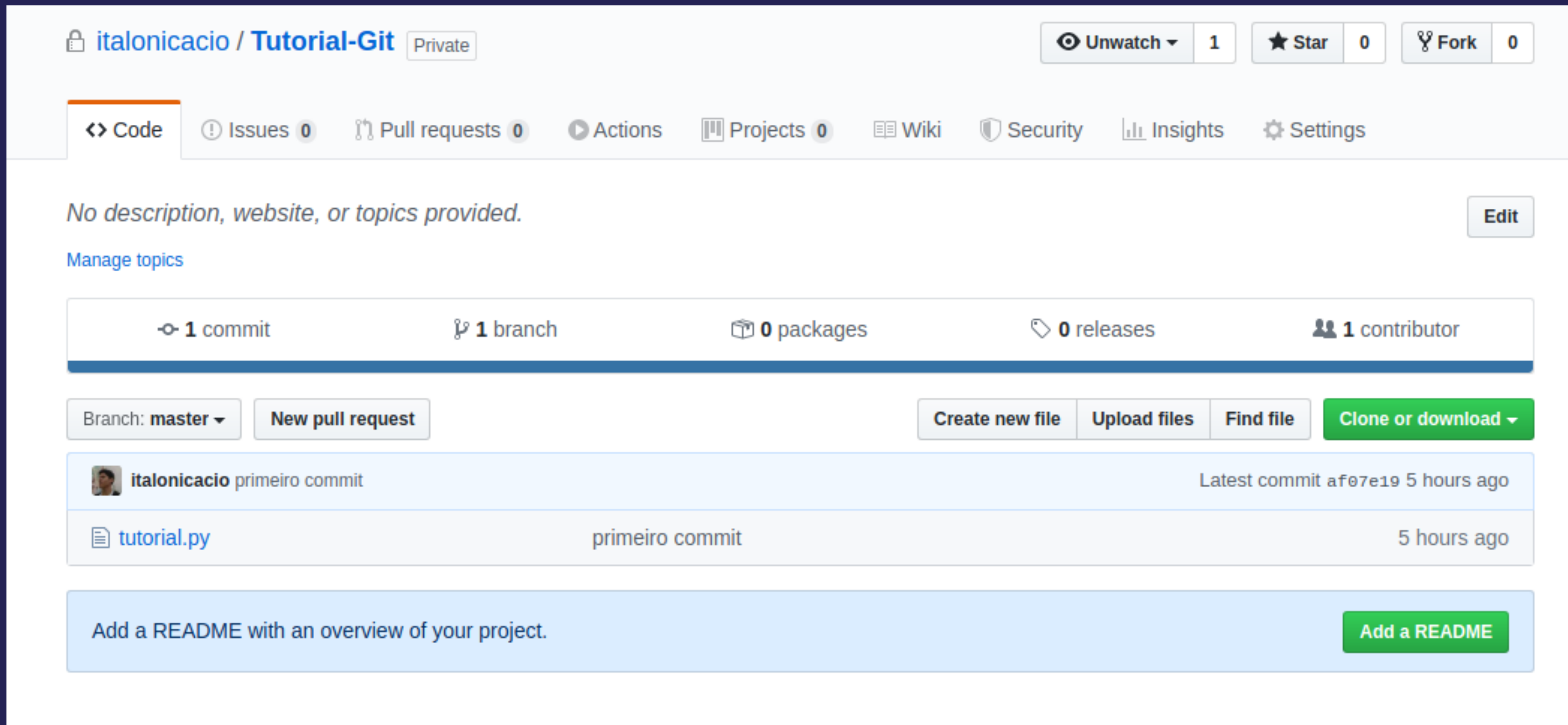
Pronto agora está tudo certo para subir para o GitHub.

```
[italo@7K tutorial_git]$ git push origin master
```

A grosso modo esse comando está adicionando os arquivos na branch master com esse comando, você pode usar esse comando trocando a master por outra branch para adicionar atualizar os arquivos na branch escolhida.

Repositorio Do GitHub

Apos essa serie de comandos se você olhar o repositório no GitHub vai está com o arquivo `tutorial.py`.



The screenshot shows a GitHub repository page for 'italonicacio / Tutorial-Git'. The repository is private and has 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. The 'Code' tab is selected, showing the 'master' branch. A file named 'tutorial.py' is listed as the latest commit, made 5 hours ago. A button 'Add a README' is visible at the bottom.

italonicacio / Tutorial-Git Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

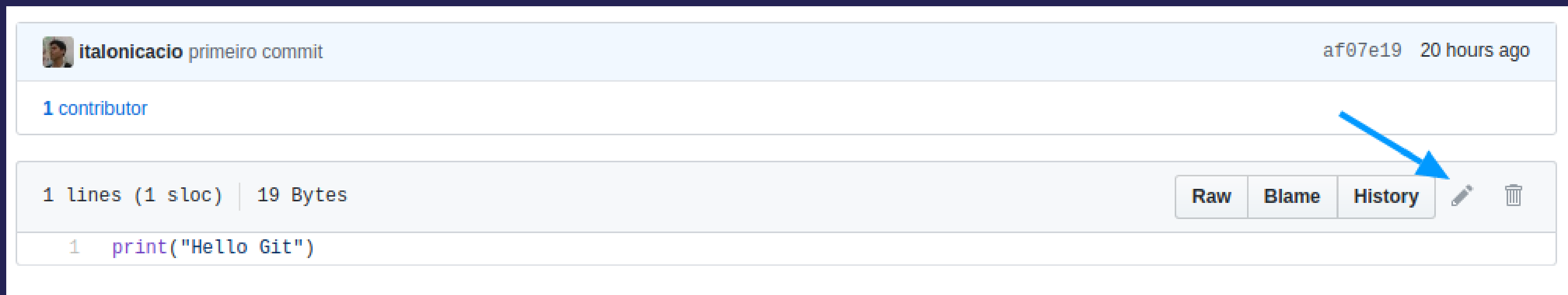
italonicacio primeiro commit Latest commit af07e19 5 hours ago

tutorial.py primeiro commit 5 hours ago

Add a README with an overview of your project. Add a README

Repositorio Do GitHub

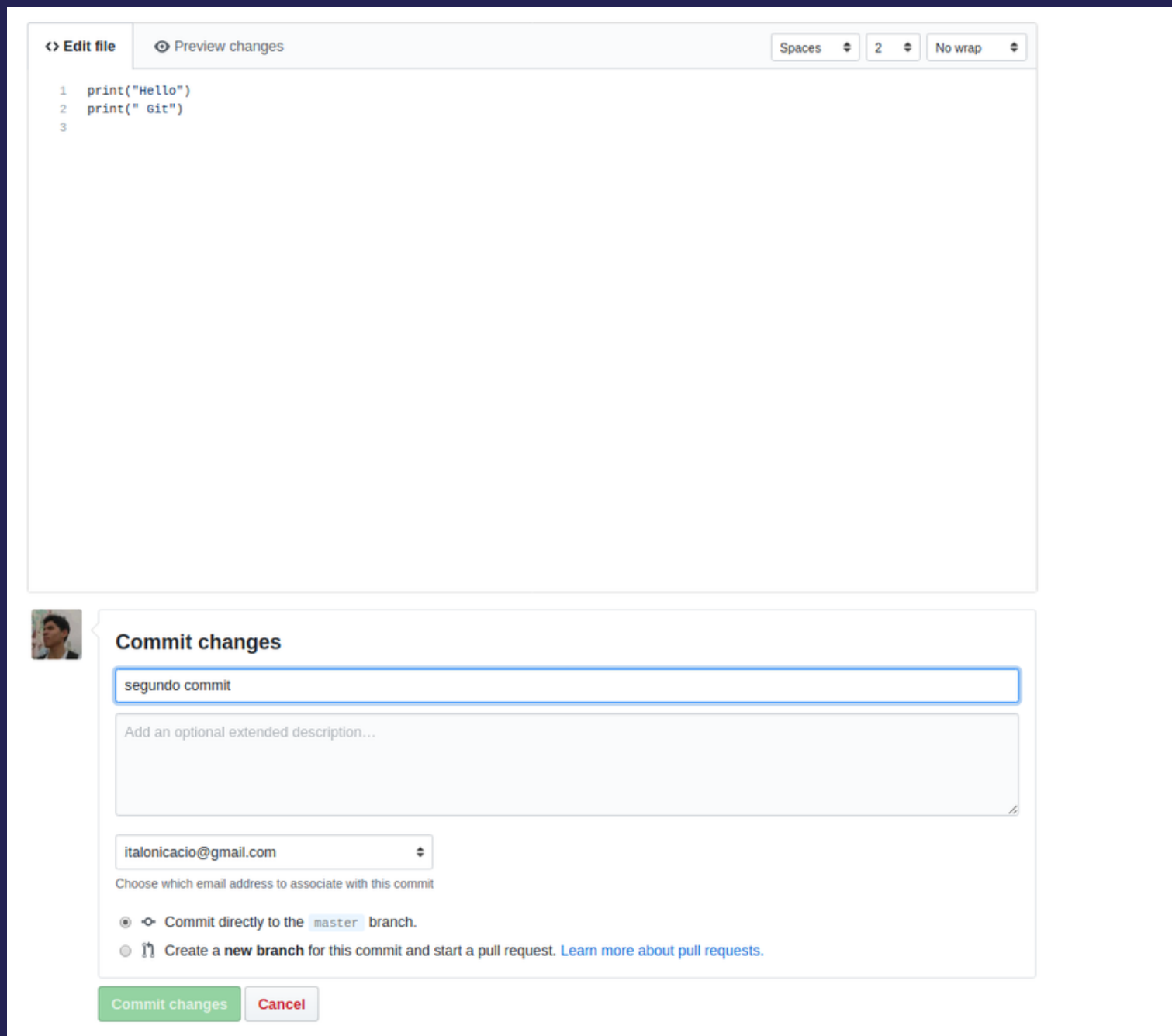
Modificando o arquivo no GitHub, apenas clique no lapiz que fica do lado da lixeira.



Apos fazer isso aparecerá a seguinte tela, nessa tela você pode modificar o arquivo e fazer commits no próprio Github, em alguns casos pode ajudar agilizar bastante essa interface.

Repositorio Do GitHub

Modifique o arquivo da forma que desejar, apos modificar basta, digitar a descrição do commit e apertar o botão verde.



The screenshot displays the GitHub web interface for editing a file. At the top, there are two tabs: "Edit file" (active) and "Preview changes". To the right of these tabs are controls for "Spaces" (set to 2) and "No wrap". The main area shows a code editor with the following content:

```
1 print("Hello")
2 print(" Git")
3
```

Below the editor is a "Commit changes" dialog. It includes a profile picture placeholder, a title "Commit changes", and a text input field containing "segundo commit". Below this is a larger text area for an optional extended description. A dropdown menu shows the email address "italonicacio@gmail.com". Below the dropdown, it says "Choose which email address to associate with this commit". There are two radio button options: "Commit directly to the master branch." (selected) and "Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)". At the bottom of the dialog are two buttons: "Commit changes" (green) and "Cancel" (red).

Repositorio Do GitHub

Bom isso foi necessário para mostrar como sincronizar o repositório Git local com o remoto, se você tentar executar o comando `git push` novamente, ocorrerá a seguinte mensagem de erro, basicamente ela diz que seu repositório local está desatualizado em relação ao repositório remoto.

```
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/italonicacio/Tutorial-Git.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Agora para sincronizar basta executar o comando `git pull`, ele irá sincronizar e mostrar o foi modificado no repositório remoto em relação ao seu repositório local.

Repositorio Do GitHub

Após executar o comando `git pull`, a seguinte mensagem irá aparecer.

```
[italo@7K tutorial_git]$ git pull origin master
Username for 'https://github.com': 
Password for 'https://[redacted]@github.com': 
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 633 bytes | 18.00 KiB/s, done.
From https://github.com/italonicacio/Tutorial-Git
 * branch                master       -> FETCH_HEAD
   af07e19..5890b1d      master       -> origin/master
Updating af07e19..5890b1d
Fast-forward
 tutorial.py | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Se você está trabalhando com alguém na mesma branch, em alguns momentos talvez será necessário executar esse comando.

Branchs

Agora falaremos sobre branchs, as branchs é fundamental para um projeto, na sua branch master, seria onde fica a sua aplicação funcionando e em uma ramificação ou branch, você pode mexer no código sem alterar a aplicação, com isso você pode fazer atualizações, implementações de novos sistemas e etc.

Para criarmos nossa primeira branch basta executar:

```
[italo@7K tutorial_git]$ git branch primeira_branch
```

Esse comando cria uma nova branch a partir da branch atual. Para trocar de branch basta executar o seguinte comando.

```
[italo@7K tutorial_git]$ git checkout primeira_branch  
Switched to branch 'primeira_branch'
```


Branchs

Vamos criar um novo arquivo.

```
[italo@7K tutorial_git]$ >primeira_branch.py
```

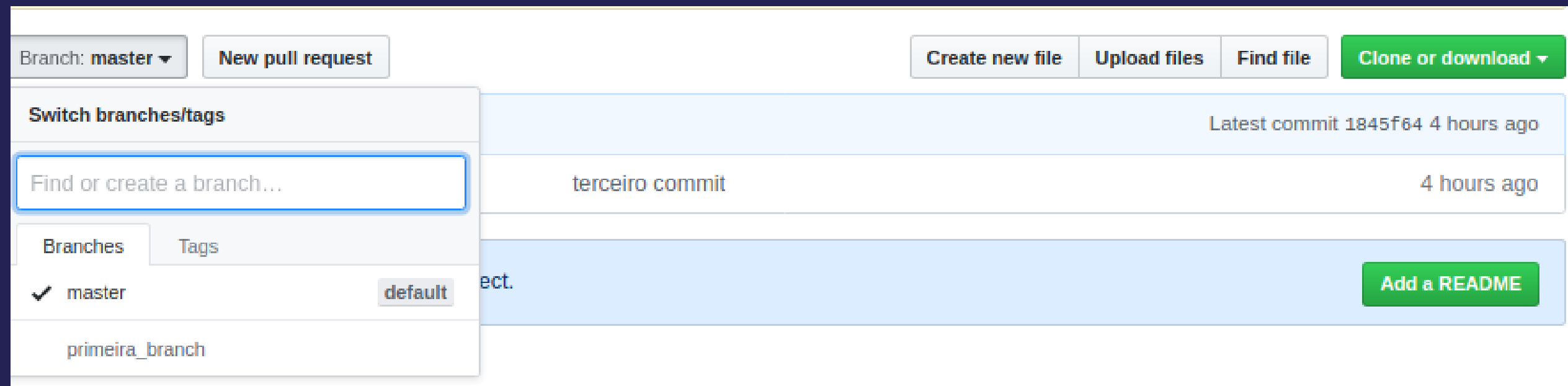
Adicione a branch atual o novo arquivo e depois faça um push.

```
[italo@7K tutorial_git]$ git add primeira_branch.py
[italo@7K tutorial_git]$ git commit -m "primeiro commit da nova branch"
[primeira_branch 330eda0] primeiro commit da nova branch
1 file changed, 1 insertion(+)
create mode 100644 primeira_branch.py
[italo@7K tutorial_git]$ git push origin primeira_branch
```

Observe que você precisa especificar qual a branch que você quer dar o push.

Branchs

Agora vamos preparar o processo de merge, que é juntar as duas branches, se você olhar seu GitHub ele estará listando as duas branches.



As alterações feitas na nova branch, não influencia na master, se você está em algum projeto, ou algum projeto pessoal, basta você fazer o merge e pronto, você juntou duas branches.


Branchs

Para fazer o merge da nova branch com a master, na branch master você deve executar o seguinte comando.



```
[italo@7K tutorial_git]$ git merge primeira_branch
Updating 1845f64..330eda0
Fast-forward
 primeira_branch.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 primeira_branch.py
```

Em seguida faça um push na master e verifique as alterações no seu GitHub.

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

 **italonicacio** primeiro commit da nova branch

Latest commit 330eda0 29 minutes ago

 primeira_branch.py	primeiro commit da nova branch	29 minutes ago
 tutorial.py	terceiro commit	4 hours ago

Add a README with an overview of your project.

Add a README

Boas Práticas Com O Git

Nessa seção será listado algumas boas praticas para se usar no Git.

- 1. Se errar a mensagem de um commit, faça a correção do mesmo, não faça outro commit por cima corrigindo o anterior.**
- 2. Limite a mensagem do commit para 50 caracteres.**
- 3. Nunca guarde arquivos binários nos seus projetos, o Git foi feito para versionar código.**
- 4. Crie outras branches para o seu projeto, não desenvolva na master.**
- 5. As mensagens de commit devem ser expressadas de forma a melhor definir o estado atual do projeto naquele ponto.**

Referências

- <https://pt.wikipedia.org/wiki/GitHub>
- <https://pt.wikipedia.org/wiki/Git>
- <https://git-scm.com>
- <https://githowto.com/pt-BR>
- <https://blog.locaweb.com.br/desenvolvedores/6-boas-praticas-de-git/>
- <http://blog.victorpre.com/git-melhores-praticas/>