



Engenharia de software

Lucas Emanuel



O que é Git?

Git é um sistema de controle de versão distribuído, usado principalmente para rastrear mudanças em arquivos de código-fonte durante o desenvolvimento de software. Ele permite que várias pessoas colaborem em projetos de forma eficiente e mantém um histórico completo de todas as alterações feitas nos arquivos ao longo do tempo.

Principais conceitos do Git

Repositório (Repository): Um repositório Git é um diretório onde todos os arquivos do seu projeto são armazenados, juntamente com as informações de controle de versão.

Commit: Um commit é uma operação que salva as alterações feitas nos arquivos do seu projeto em um repositório Git. Cada commit possui uma mensagem associada que descreve as alterações realizadas.

Principais conceitos do Git

Branch: Um branch é uma ramificação do seu projeto, que permite que você trabalhe em novas funcionalidades ou correções de bugs sem interferir no branch principal (geralmente chamado de master ou main).

Merge: Merge é o processo de combinar as alterações de um branch com outro. Isso é geralmente feito para incorporar as alterações de uma feature branch de volta para o branch principal.

O que é Github?

GitHub é uma plataforma de hospedagem de código-fonte que utiliza o Git para controle de versão. Ele oferece recursos adicionais, como controle de acesso, rastreamento de problemas, integração contínua e colaboração em equipe. O GitHub é amplamente utilizado pela comunidade de desenvolvimento de software para hospedar e colaborar em projetos de código aberto e privados.



Principais conceitos do Github

Repositório Remoto (Remote Repository): Um repositório no GitHub é um repositório Git hospedado na plataforma do GitHub. Ele serve como uma cópia centralizada do seu projeto que pode ser acessada e colaborada por outros desenvolvedores.

Fork: Um fork é uma cópia de um repositório GitHub que pode ser feita por qualquer usuário. Ele permite que você contribua para um projeto sem modificar diretamente o repositório original.



Principais conceitos do Github

Pull Request: Um pull request é uma solicitação para incorporar as alterações feitas em um fork de volta para o repositório original. Ele é usado para revisão e discussão das alterações antes de serem mescladas.

Issues: Issues são usadas para rastrear tarefas, bugs ou discussões relacionadas ao projeto. Elas são usadas para colaboração e comunicação entre os membros da equipe.

Primeiros passos com Git

Criar uma conta

Visite <https://github.com/> para criar a sua conta no Github

Primeiros passos com Git

Criar uma conta

Visite <https://github.com/> para criar a sua conta no Github

Primeiros passos com Git

Abra o terminal (prompt de comando) da sua máquina e digite: `git --version` . Caso o git estiver instalado, ele vai retornar a versão do git que está em sua máquina.

Primeiros passos com Git

Abra o terminal (prompt de comando) da sua máquina e digite: `git config --global user.name .` Para verificar o **nome** global do usuário git associado à máquina.

digite: `git config --global user.email .` Para verificar o **e-mail** global do usuário git associado à máquina.



Primeiros passos com Git

Abra o terminal (prompt de comando) da sua máquina e digite: `git config --global user.name "Seu Nome" .` Para modificar o **nome** global do usuário git associado à máquina. Substitua "Seu Nome" pelo seu nome de usuário desejado.

digite: `git config --global user.email "seu@email.com" .` Para modificar o **e-mail** global do usuário git associado à máquina. Substitua "seu@email.com" pelo seu endereço de e-mail desejado.



Primeiros passos com Git

Inicializando um repositório: Crie uma pasta no seu computador e entre nela.

Por exemplo no terminal:

```
mkdir aulagit
```

```
cd aulagit
```

Primeiros passos com Git

Inicializando um repositório: Abra o terminal dentro da pasta criada e digite:

```
git init
```

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git init
Initialized empty Git repository in C:/Users/User/Desktop/ELO-ADS/EngSoftware/aulagit/.git/
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>
```

Primeiros passos com Git

Inicializando um repositório: Foi criado um diretório (pasta) oculto chamado .git. Para vê-lo digite:

`dir /a`

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>dir /a
O volume na unidade C não tem nome.
O Número de Série do Volume é 5001-C57A

Pasta de C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit

15/04/2024  11:33    <DIR>          .
15/04/2024  11:33    <DIR>          ..
15/04/2024  11:33    <DIR>          .git
               0 arquivo(s)                0 bytes
               3 pasta(s)  77.388.333.056 bytes disponíveis
```


Primeiros passos com Git

Fluxo de trabalho com Git: Foi criado um diretório (pasta) oculto chamado .git. Para vê-lo digite:

```
dir /a
```

Para verificar o status atual do repositório: `git status`

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git status
On branch master

No commits yet

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

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

Primeiros passos com Git

Fluxo de trabalho com Git: Precisamos adicionar os arquivos criados para a área de stagin. Utilizaremos o comando: `git add <nome do arquivo>` ou `git add .` Com esse '.' no final para adicionar todos os novos arquivos.

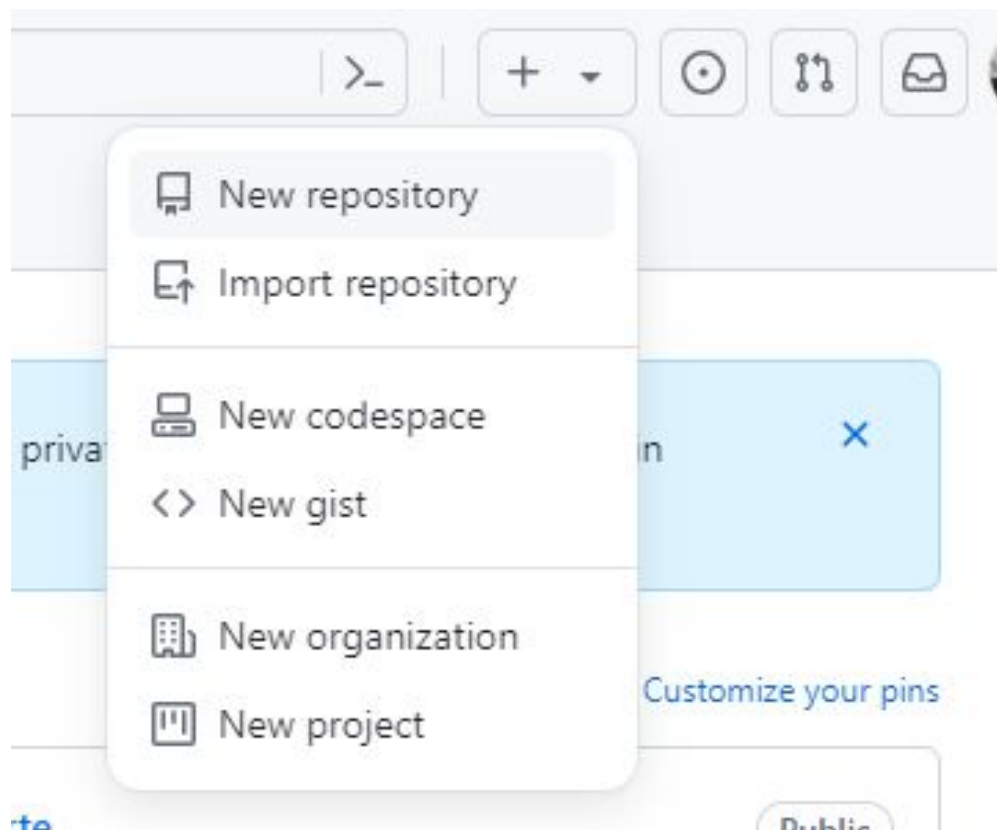
```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git add .  
  
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
t (use "git rm --cached <file>..." to unstage)  
    new file:   teste.py
```

Primeiros passos com Git

Fluxo de trabalho com Git: Precisamos armazenar no repositório git os arquivos adicionados. Utilizaremos o comando `git commit -m "nome do commit"`

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git commit -m "primeiro commit"
[master (root-commit) 08ca5cf] primeiro commit
1 file changed, 14 insertions(+)
 create mode 100644 teste.py
```

Criando um repositório no Github



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template

No template

Start your repository with a template repository's contents.

Owner *



lluk4z

Repository name *

aula-git

aula-git is available.

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

Description (optional)



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.

Initialize this repository with:

Criando um repositório no Github

☐ 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.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a private repository in your personal account.

Create repository



Criando um repositório no Github

```
git remote add origin git@github.com:lluk4z/aula-git.git
```

```
git branch -M master
```

```
git push -u origin master
```



Criando um repositório no Github

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git remote add origin git@github.com:lluk4z/aula-git.git

C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git branch -M master

C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git push -u origin master
Enter passphrase for key '/c/Users/User/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 343 bytes | 114.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:lluk4z/aula-git.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```


Caso não tenhamos um repositório já criado localmente

```
echo "# aula-git" >> README.md
```

```
git init
```

```
git add README.md
```

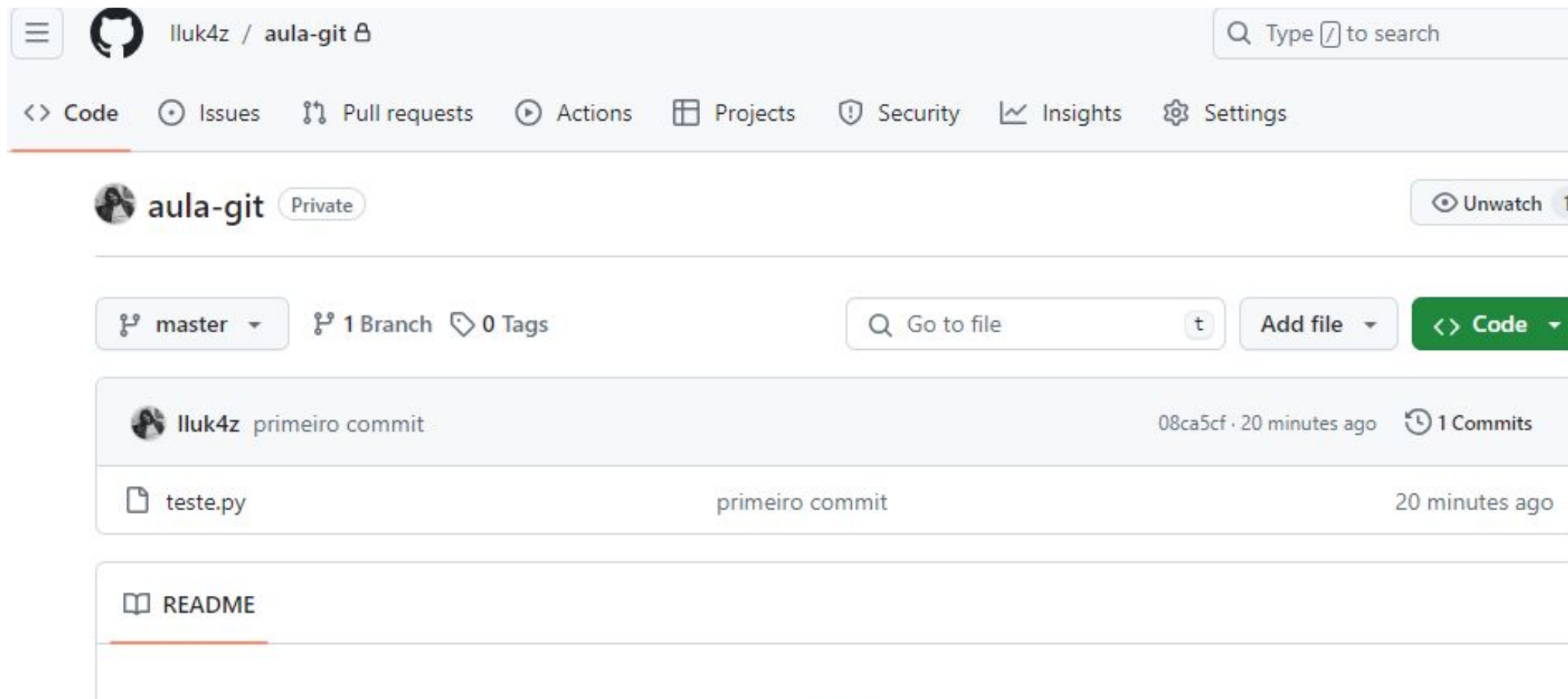
```
git commit -m "first commit"
```

```
git branch -M master
```

```
git remote add origin git@github.com:lluk4z/aula-git.git
```

```
git push -u origin master
```

Criando um repositório no Github



Enviando mudanças para o repositório no Github

Se caso fizermos alguma mudança no repositório, agora podemos enviá-los ao github:

```
git add.
```

```
git commit -m "nome do commit"
```

```
git push origin <nome da branch> ou só git push
```

Enviando mudanças para o repositório no Github

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   teste.py

no changes added to commit (use "git add" and/or "git commit -a")
```

Enviando mudanças para o repositório no Github

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git add .

C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   teste.py

C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git commit -m "Mudanças"
[master b907745] Mudanças
 1 file changed, 5 insertions(+), 1 deletion(-)

C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

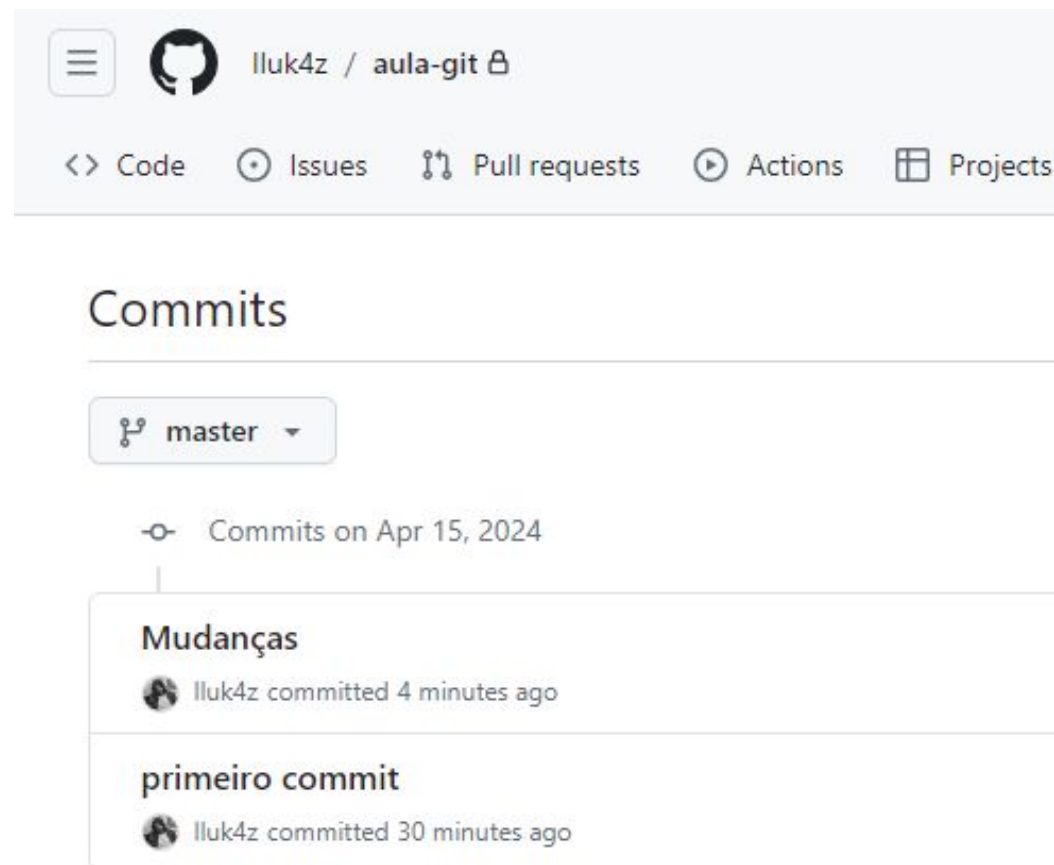
nothing to commit, working tree clean
```


Enviando mudanças para o repositório no Github

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git branch
* master

C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git push origin master
Enter passphrase for key '/c/Users/User/.ssh/id_rsa':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 310 bytes | 103.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:lluk4z/aula-git.git
   08ca5cf..b907745  master -> master
```

Enviando mudanças para o repositório no Github



Recuperando alterações do repositório pelo Github

Caso algum outro membro da equipe tenha alterado o repositório no Github, como podemos fazer para enviar essas alterações para o nosso repositório local?

Utilizaremos o comando `git pull`

Recuperando alterações do repositório pelo Github

```
C:\Users\User\Desktop\ELO-ADS\EngSoftware\aulagit>git pull
Enter passphrase for key '/c/Users/User/.ssh/id_rsa':
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 2.02 KiB | 57.00 KiB/s, done.
From github.com:lluk4z/aula-git
   b907745..25ed8b1  master    -> origin/master
Updating b907745..25ed8b1
Fast-forward
 README.md | 23 ++++++
1 file changed, 23 insertions(+)
create mode 100644 README.md
```



Exercícios

- Crie um diretório local e transforme ele num repositório git
- Dentro dessa pasta crie um arquivo python chamado exercicios.py
- Crie uma calculadora simples em Python que permita aos usuários realizar operações básicas de adição, subtração, multiplicação e divisão. Os usuários devem ser capazes de inserir os números e a operação desejada através de entradas do teclado.



Exercícios

- Adicione as alterações do arquivo `exercicios.py` para a área de stage com `git add .`
- Comite as alterações no repositório
- Crie um repositório remoto no github
- Envie o repositório local da sua máquina para o repositório remoto no github