Git & GitHub

Introdução

Prática de Laboratório

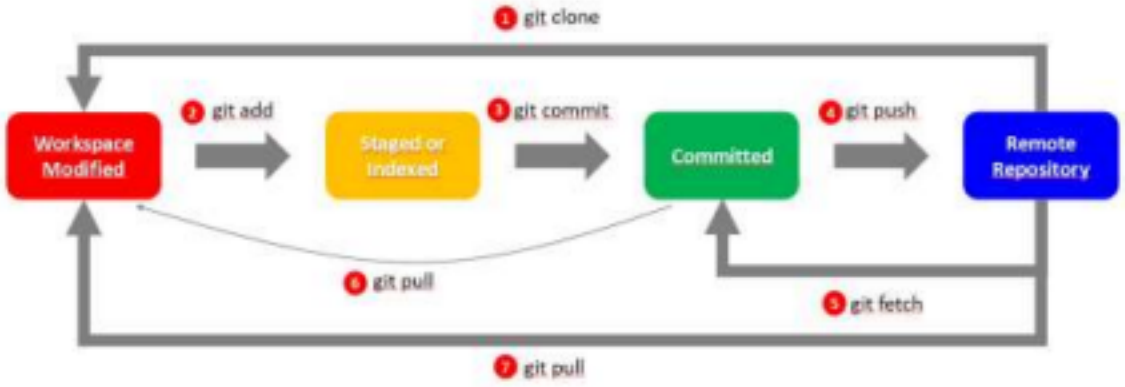
Wesley Dias Maciel / Thiérs Hofman

2023

Exercício 01

Objetivo:

Sequência que será executada nesta prática:



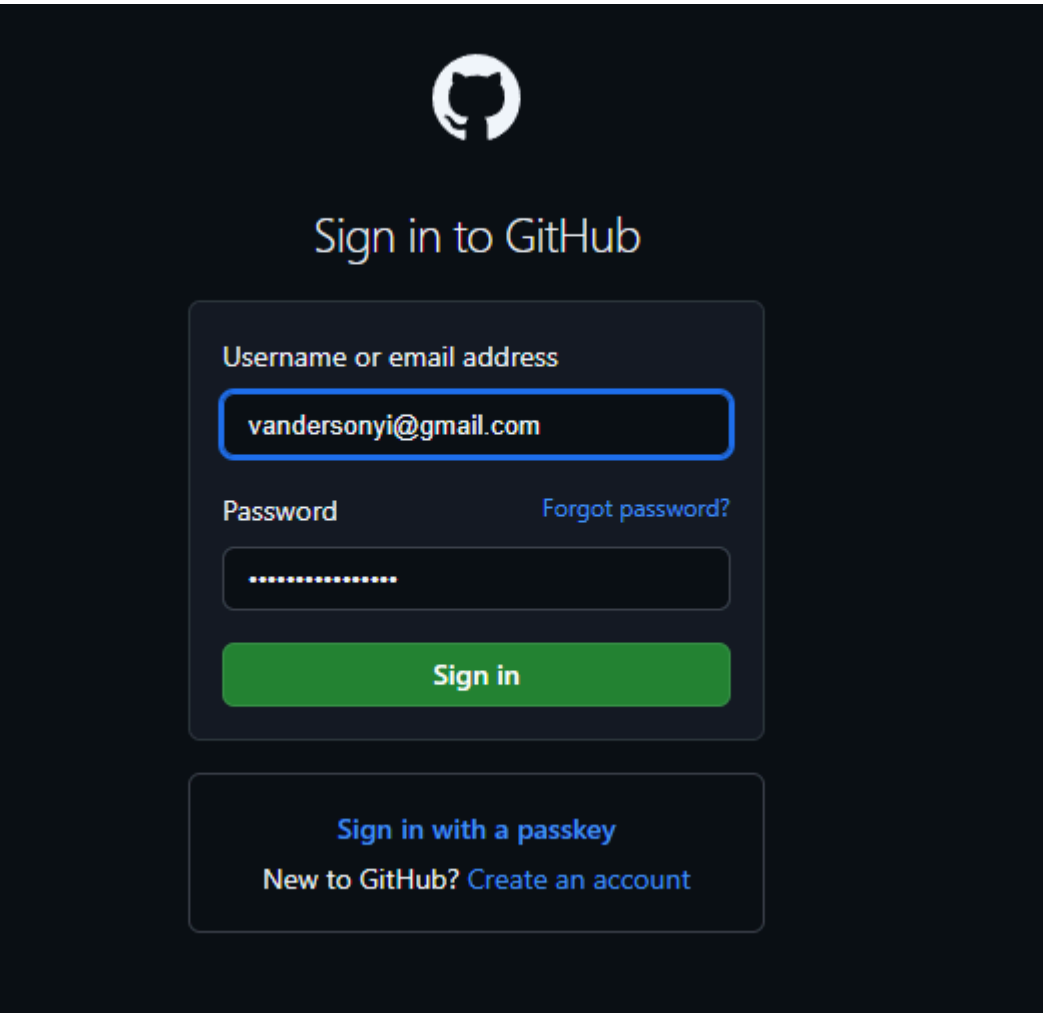
Aluno: vanderson henrique da silva correia filho

Ra: 4231922369

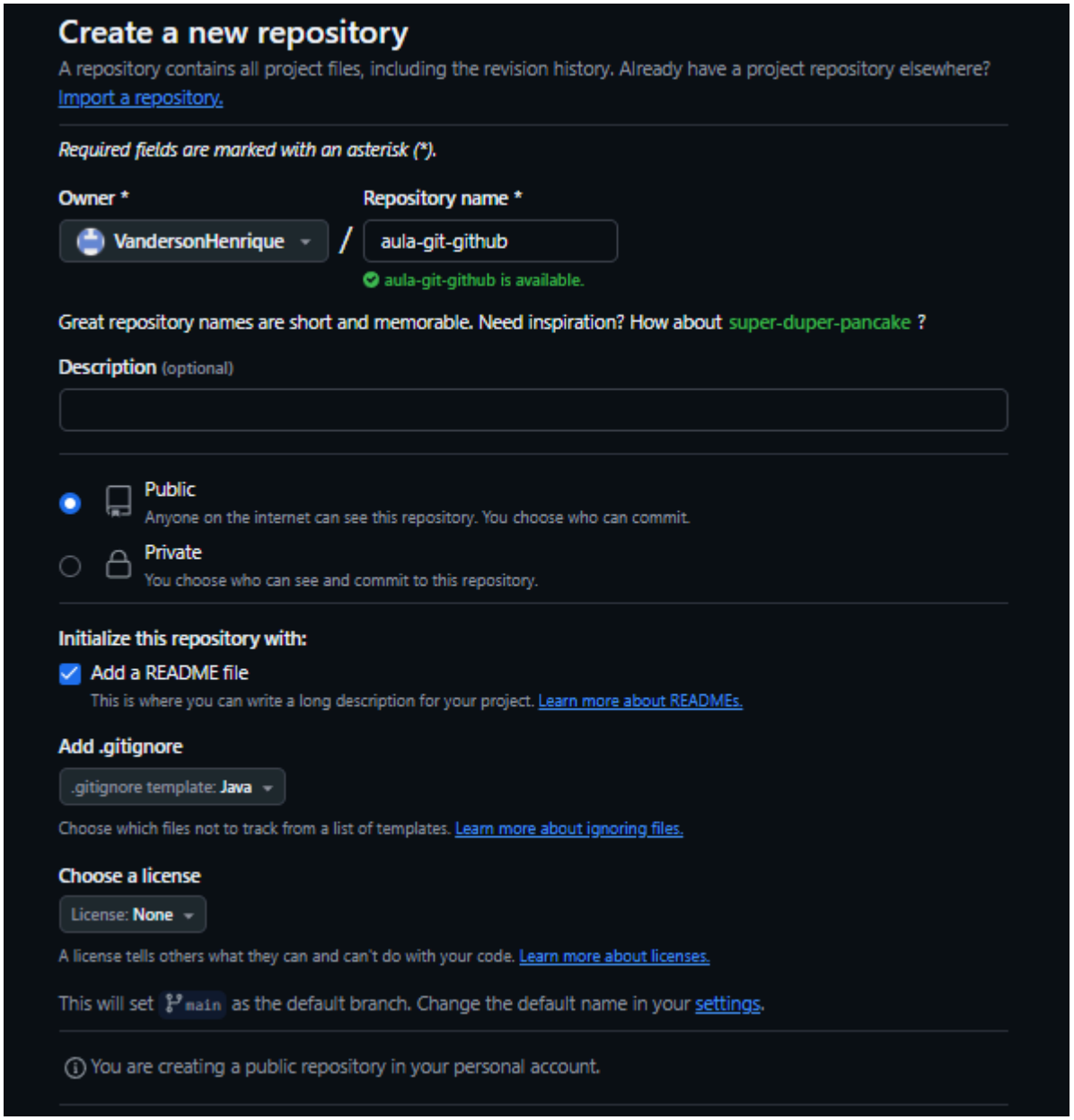
GitHub: criar repositório remoto.

1) Página do GitHub: https://github.com/.

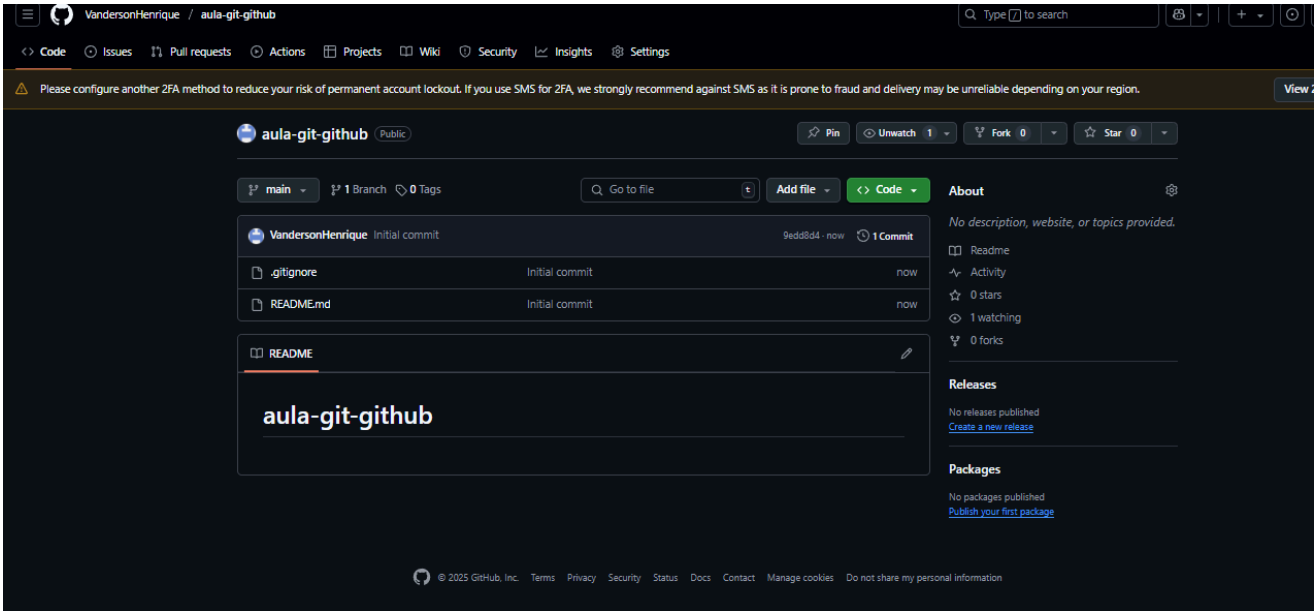
2) Entrar: https://github.com/login.



3) Página de criação de novo repositório: https://github.com/new. Crie um novo repositório público com nome “aula-git-github” e adicione os arquivos “README” e “.gitignore” (“template Java”), como apresentado abaixo.



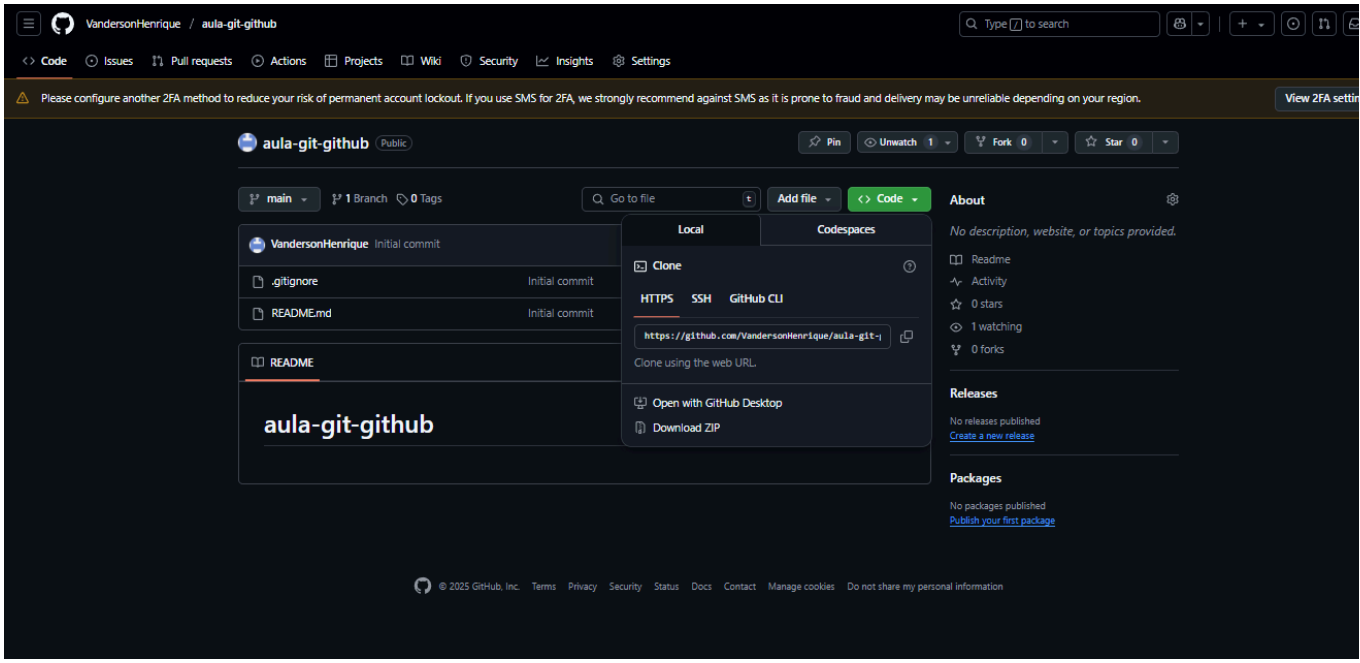
4) Página inicial do repositório: https://github.com/wesleydiasmaciel/aula-git-github.



Exercício 02

Git: clonar repositório remoto.

1) Copie o link do seu repositório remoto:



2) Num diretório local de sua máquina, clone o repositório remoto:

$ git clone https://github.com/wesleydiasmaciel/aula-git-github.git

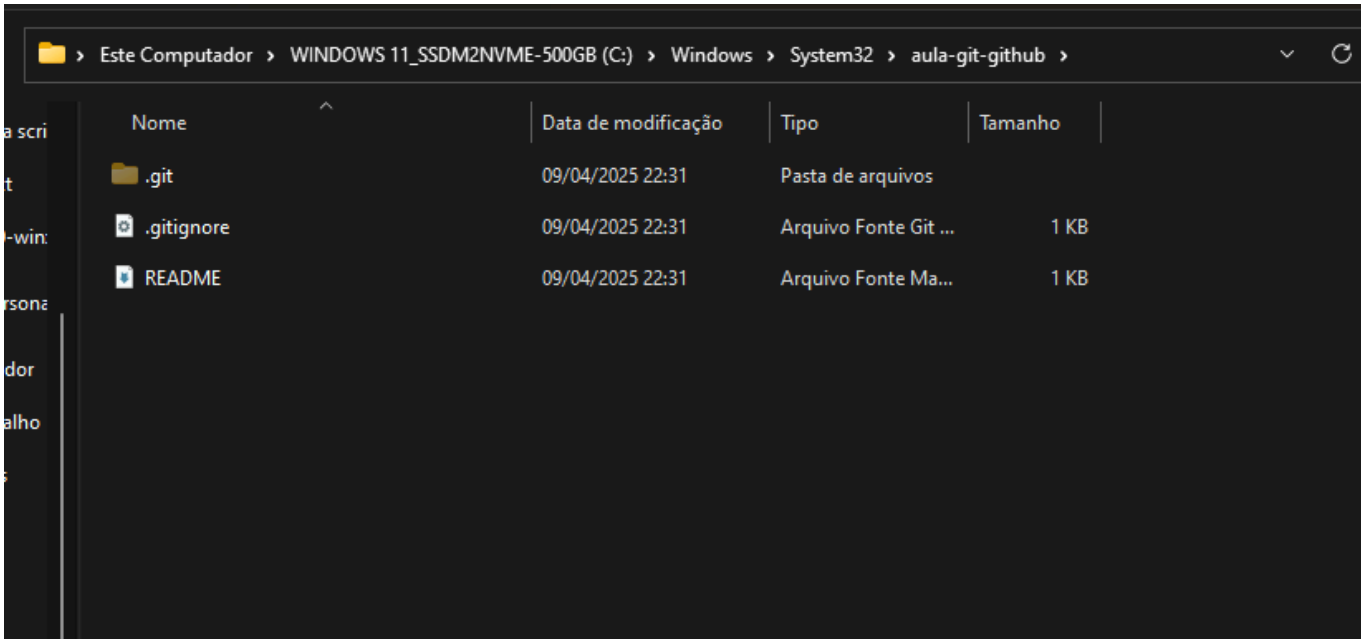
```
C:\Windows\system32>git clone https://github.com/VandersonHenrique/aula-git-github.git
Cloning into 'aula-git-github'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

3) Mude (entre) no novo diretório local de sua máquina que contém o clone do repositório remoto:

$ cd aula-git-github

```
C:\Windows\system32>cd aula-git-github

C:\Windows\System32\aula-git-github>
```

4) Observar o conteúdo do diretório criado.



git status

5) Reportar o estado do repositório.

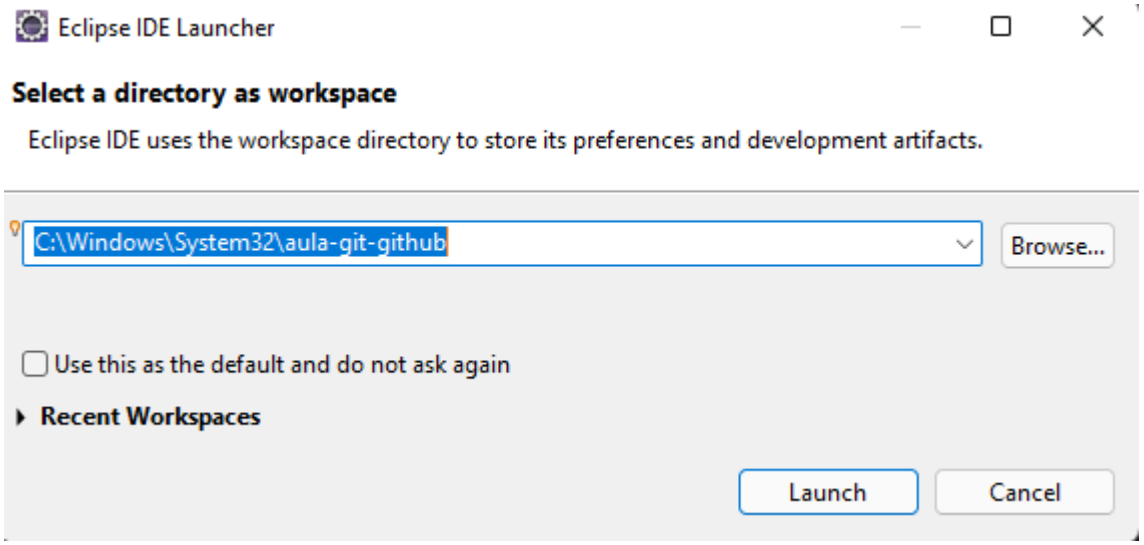
$ git status]

```
C:\Windows\System32\aula-git-github>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

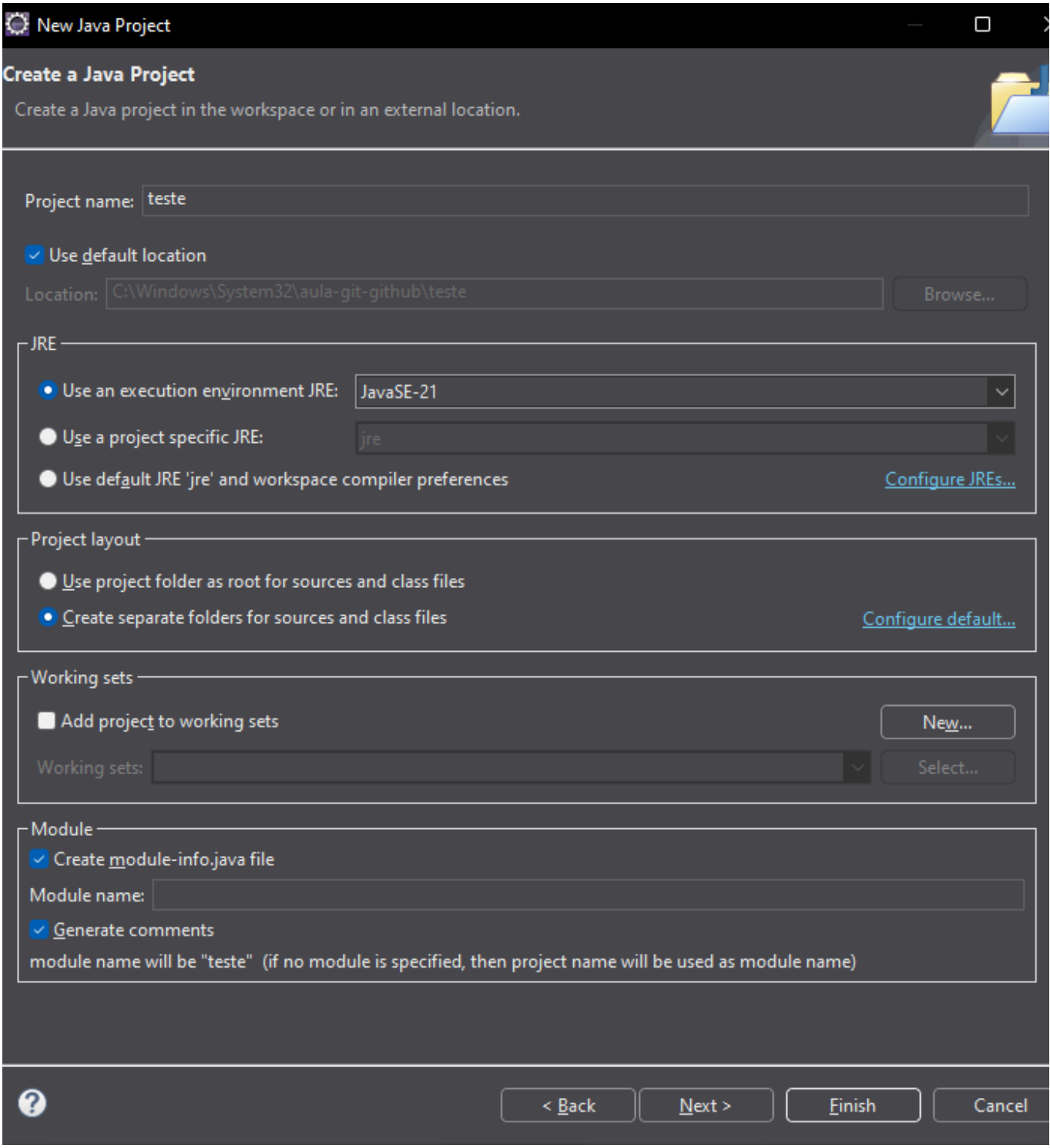
Exercício 03

Git: adicionar arquivo a um repositório local e enviar para o repositório remoto.

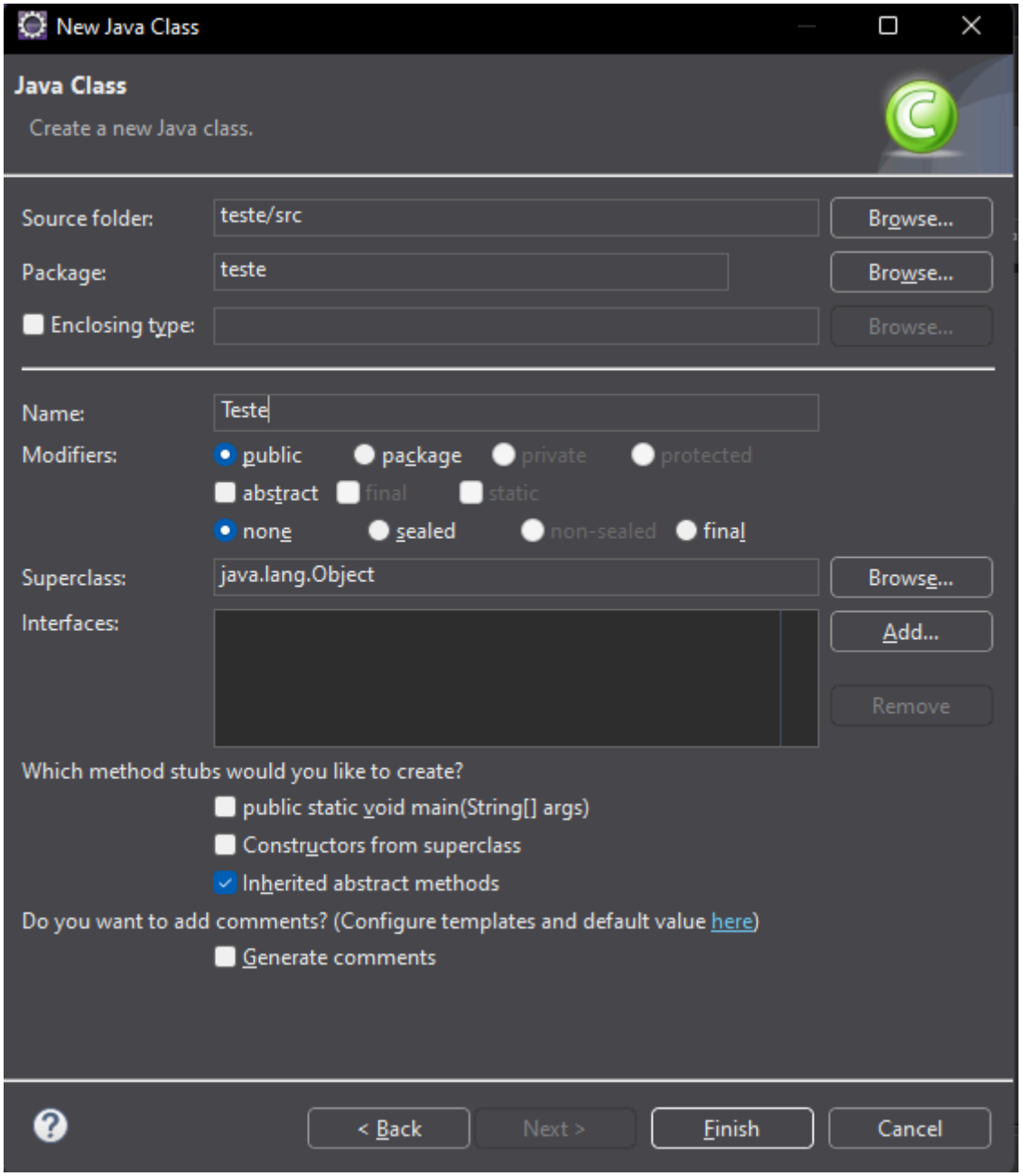
1) No IDE Eclipse, crie o “workspace” dentro do diretório do repositório local “aula-git github”.



2) No “wokspace” do IDE Eclipse, crie um novo projeto Java com nome “teste”:



3) No projeto “teste”, crie a classe “Teste” abaixo:



git status

4) Reportar o estado do repositório.

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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: .gitignore

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

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

```
C:\Windows\System32\aula-git-github> git status
On branch main
Your branch is up to date with 'origin/main'.

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:   .gitignore

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

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

git add

5) Adicionar o arquivo na “staging area”.

$ git add .

```
C:\Windows\System32\aula-git-github>git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'teste/.gitignore', LF will be replaced by CRLF the next time Git touches
```

git status

6) Reportar o estado do repositório.

$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified: .gitignore
new file: teste/.classpath
new file: teste/.gitignore
new file: teste/.project
new file: teste/.settings/org.eclipse.jdt.core.prefs
new file: teste/src/teste/Teste.java

```
C:\Windows\System32\aula-git-github>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore
        new file:   teste/.classpath
        new file:   teste/.gitignore
        new file:   teste/.project
        new file:   teste/.settings/org.eclipse.core.resources.prefs
        new file:   teste/.settings/org.eclipse.jdt.core.prefs
        new file:   teste/src/module-info.java
        new file:   teste/src/teste/Teste.java
```

git commit

Responsável por criar uma imagem, “snapshot”. Cria uma versão.

7) Criar o primeiro “commit” do projeto.

$ git commit -m "Primeiro commit do projeto: classe Java Teste vazia."

[main 3c27e6f] Primeiro commit do projeto: classe Java Teste vazia.
6 files changed, 53 insertions(+)
create mode 100644 teste/.classpath
create mode 100644 teste/.gitignore
create mode 100644 teste/.project
create mode 100644 teste/.settings/org.eclipse.jdt.core.prefs
create mode 100644 teste/src/teste/Teste.java

```
C:\Windows\System32\aula-git-github> git commit -m "Primeiro commit do projeto: classe Java Teste vazia."
[main 59b3533] Primeiro commit do projeto: classe Java Teste vazia.
 8 files changed, 55 insertions(+)
 create mode 100644 teste/.classpath
 create mode 100644 teste/.gitignore
 create mode 100644 teste/.project
 create mode 100644 teste/.settings/org.eclipse.core.resources.prefs
 create mode 100644 teste/.settings/org.eclipse.jdt.core.prefs
 create mode 100644 teste/src/module-info.java
 create mode 100644 teste/src/teste/Teste.java
```

git push

Enviar do repositório local para o repositório remoto. 8) Realizar

envio do repositório local para o repositório remoto.

$ git push

Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 1.48 KiB | 759.00 KiB/s, done.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/wesleydiasmaciel/aula-git-github.git
b5522cb..3c27e6f main -> main

```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github> git push
info: please complete authentication in your browser...
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 1.56 KiB | 532.00 KiB/s, done.
Total 14 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/VandersonHenrique/aula-git-github.git
   9edd8d4..59b3533  main -> main
```

git status

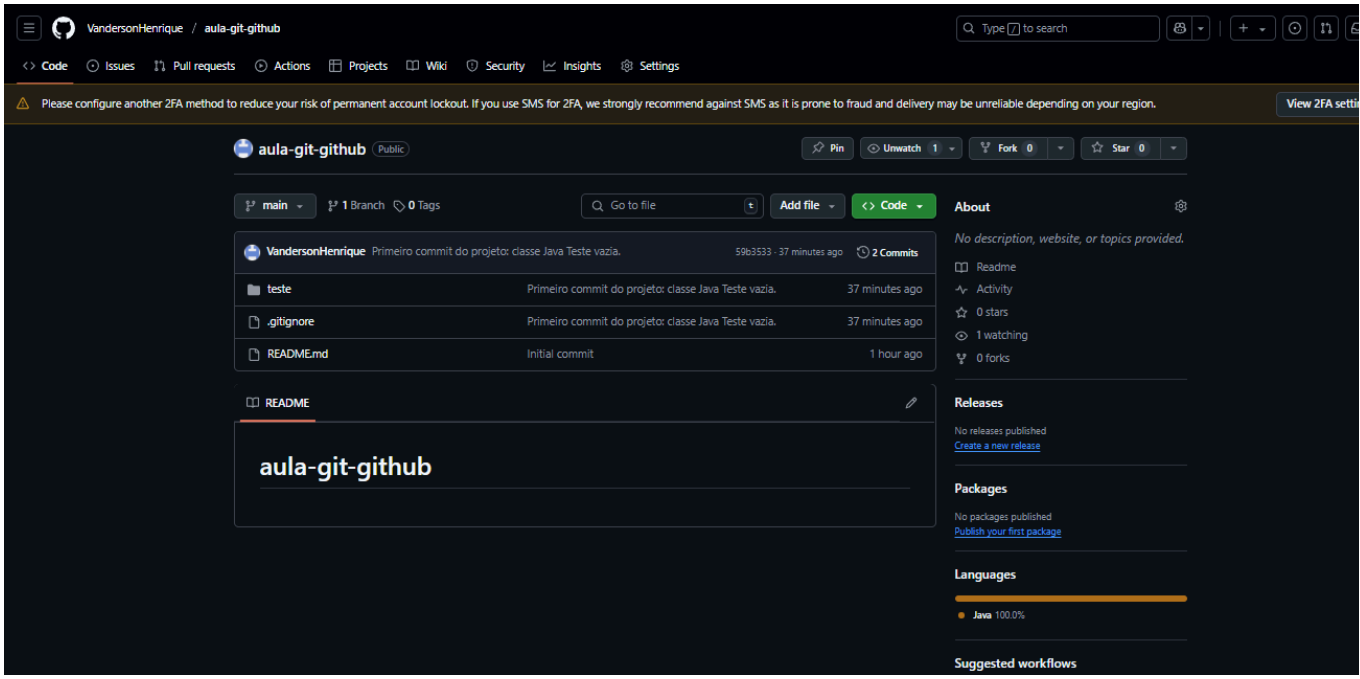
9) Reportar o estado do repositório.

$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

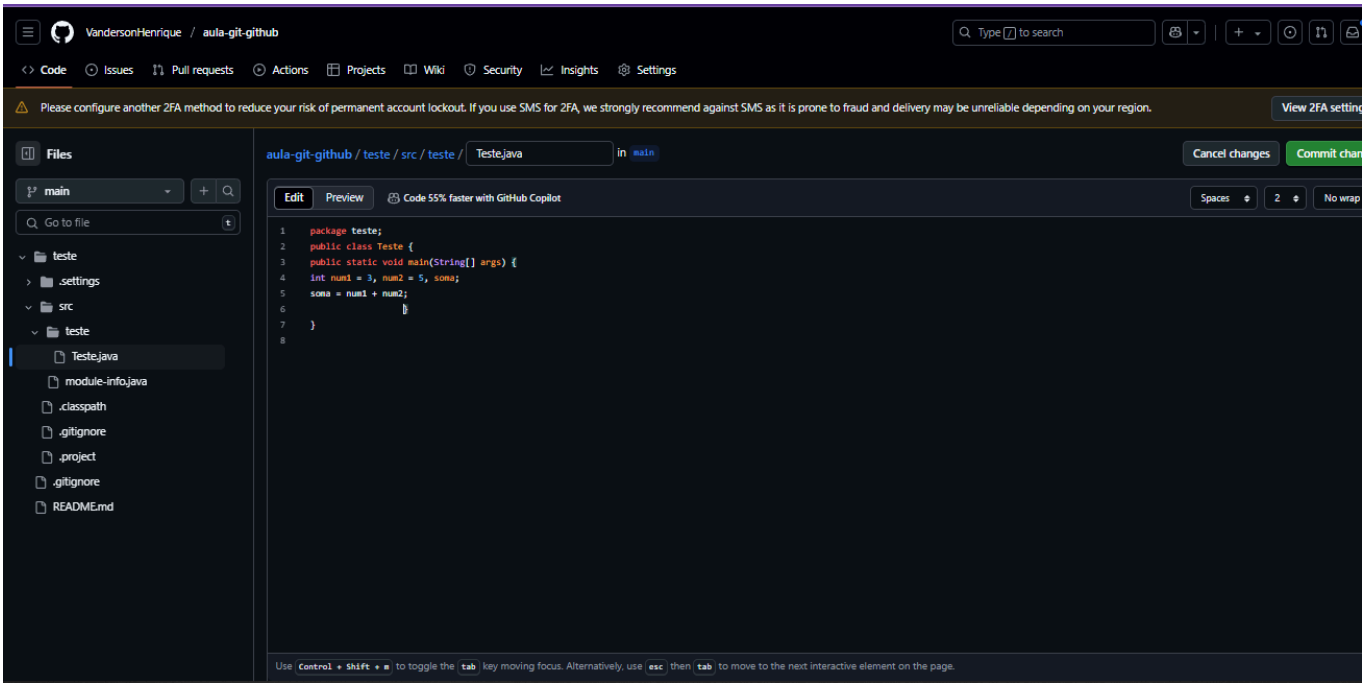
```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

10) Atualizar a página do repositório remoto no navegado e observar o conteúdo.



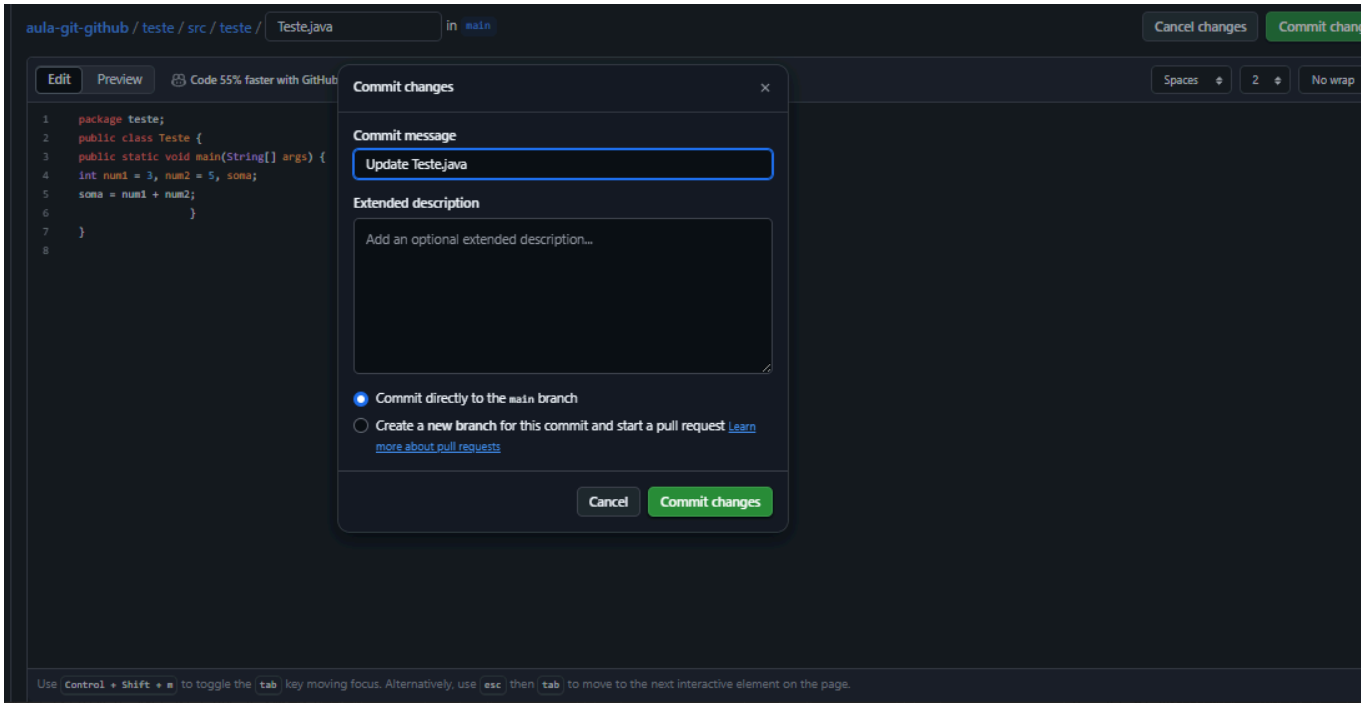
Git: alterar o diretório remoto e buscar para o repositório local sem realizar o “merge”.

1) Alterar a classe Java Teste no repositório remoto, como indicado abaixo.




```java
package teste;
public class Teste {

	public static void main(String[] args) {
		int num1 = 3, num2 = 5, soma;

		soma = num1 + num2;

	}

}
```

2) Realizar o “commit” das alterações:



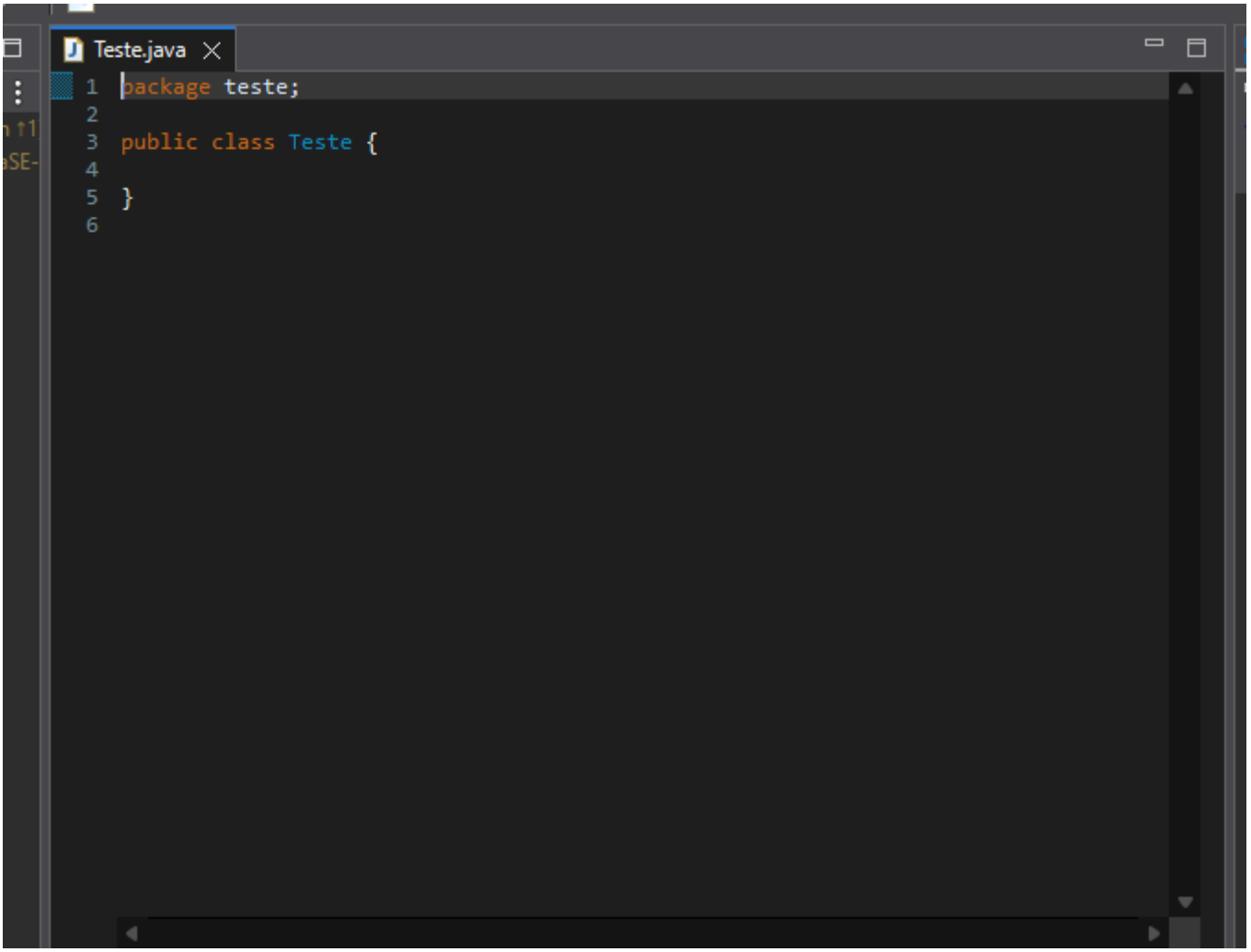
3) Buscar as alterações para o repositório local.

$ git fetch

remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 886 bytes | 10.00 KiB/s, done.
From https://github.com/wesleydiasmaciel/aula-git-github
3c27e6f..42ebfea main -> origin/main

```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github>  git fetch
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 1.18 KiB | 120.00 KiB/s, done.
From https://github.com/VandersonHenrique/aula-git-github
   59b3533..9fef3d8  main       -> origin/main
```

4) Fechar a janela da classe Java Teste no Eclise e abrir novamente. Observar que as alterações realizadas remotamente não foram aplicadas ao projeto Java.



git status

5) Reportar o estado do repositório.

$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)

nothing to commit, working tree clean

```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github> git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
```

git pull

6) Realizar o merge das alterações.

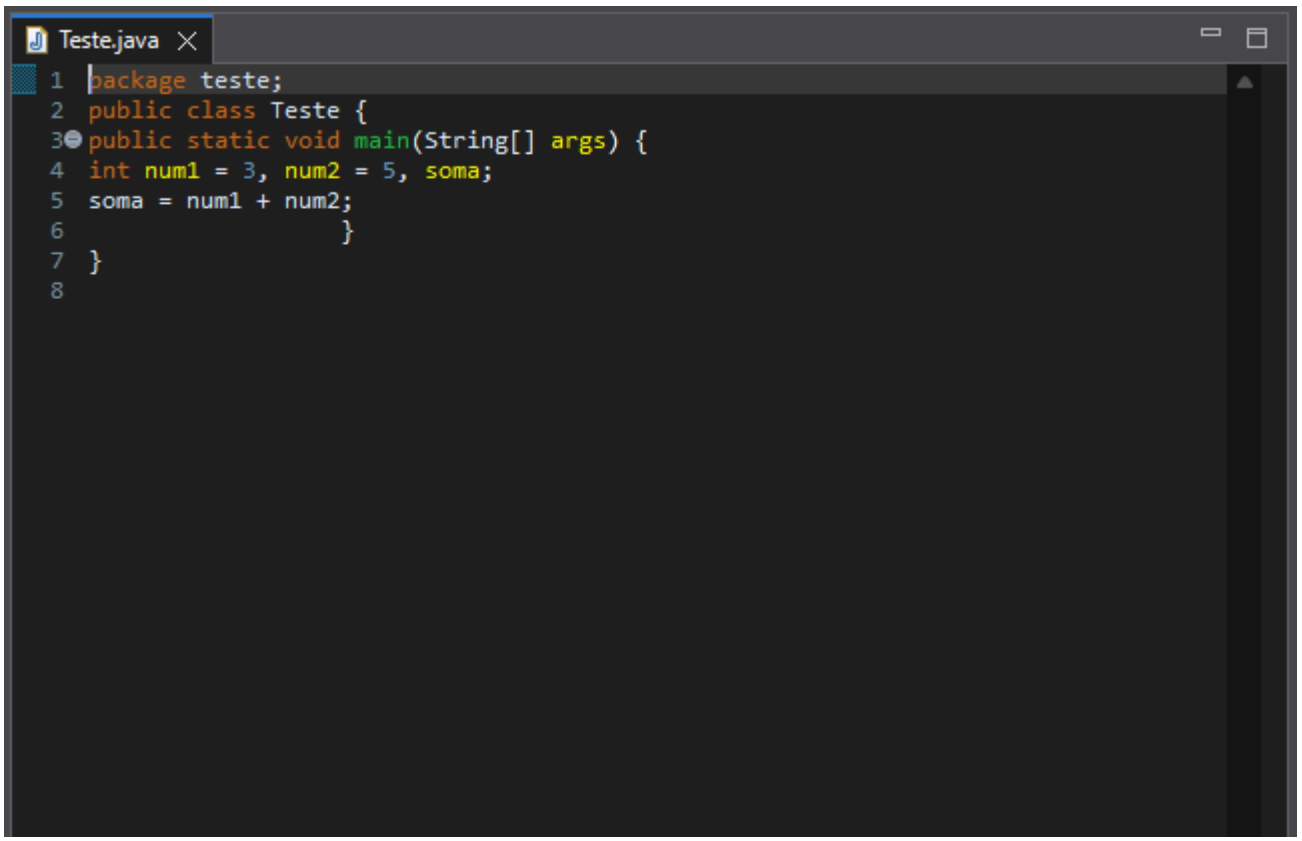
$ git pull

Updating 3c27e6f..42ebfea
Fast-forward
teste/src/teste/Teste.java | 4 +++-

1 file changed, 3 insertions(+), 1 deletion(-)

```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github> git pull
Updating 59b3533..9fef3d8
Fast-forward
 teste/src/teste/Teste.java | 12 +++++++------
 1 file changed, 7 insertions(+), 5 deletions(-)
```

7) Fechar a janela da classe Java Teste no Eclise e abrir novamente. Observar que as alterações realizadas remotamente foram aplicadas ao projeto Java.



git status

8) Reportar o estado do repositório.

$ git status
On branch main
Your branch is up to date with 'origin/main'.

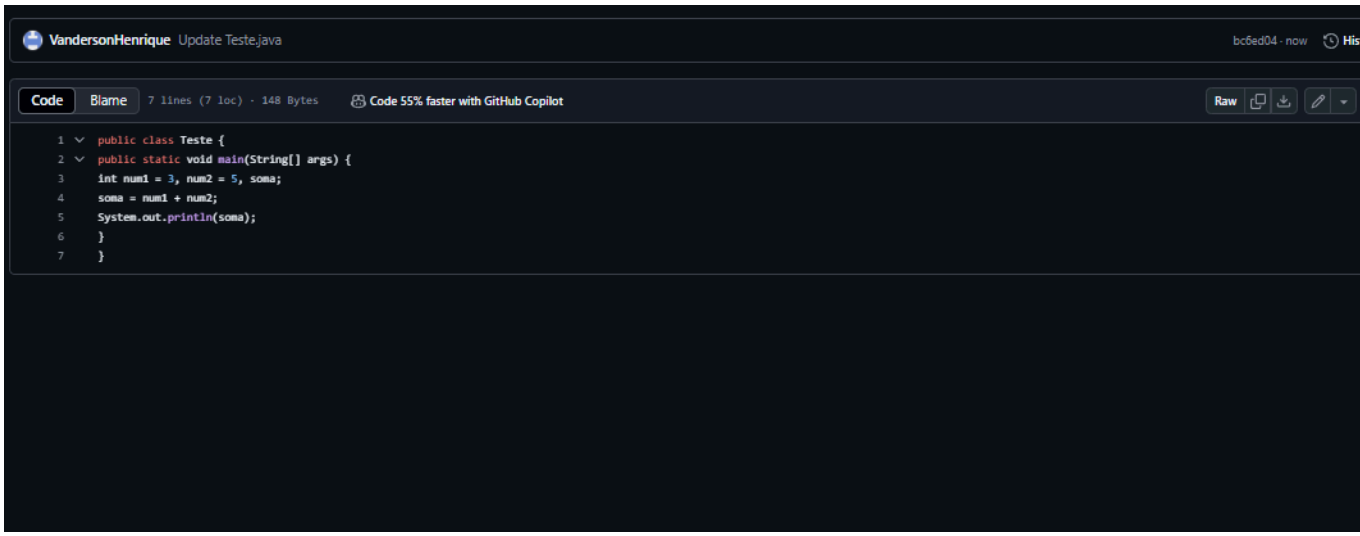
nothing to commit, working tree clean

```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Git: alterar o diretório remoto, buscar para o repositório local e realizar o “merge”.

1) Alterar a classe Java Teste no repositório remoto, como indicado abaixo.

```
package teste;
public class Teste {

	public static void main(String[] args) {
		int num1 = 3, num2 = 5, soma;

		soma = num1 + num2;

		System.out.println(soma);
	}
}
```

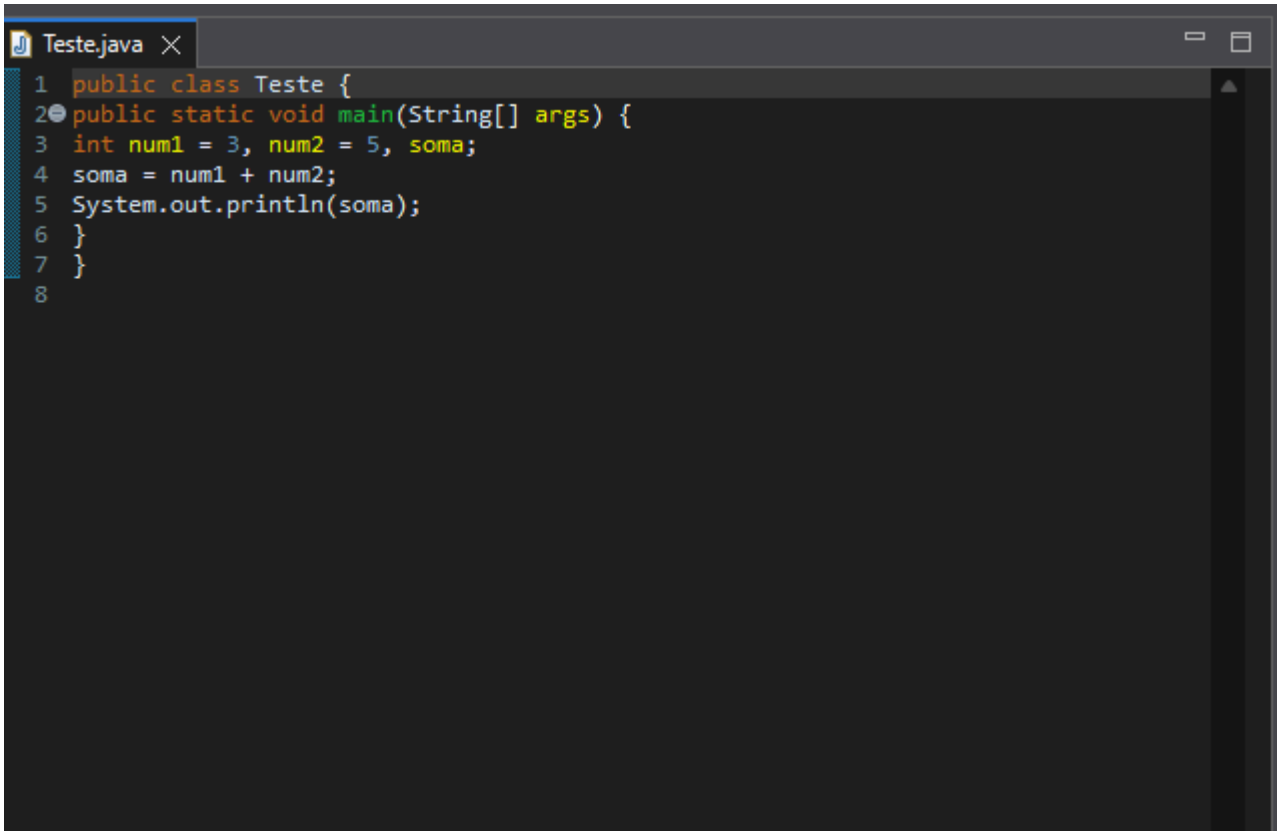


2) Buscar as alterações para o repositório local e realizar o “merge”.

```
$ git pull

remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 837 bytes | 19.00 KiB/s, done.
From https://github.com/wesleydiasmaciel/aula-git-github
 42ebfea..8477136 main -> origin/main
Updating 42ebfea..8477136
Fast-forward
 teste/src/teste/Teste.java | 2 ++
 1 file changed, 2 insertions(+)
```

```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github> git pull
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 1.13 KiB | 115.00 KiB/s, done.
From https://github.com/VandersonHenrique/aula-git-github
   9fef3d8..bc6ed04  main       -> origin/main
Updating 9fef3d8..bc6ed04
Fast-forward
 teste/src/teste/Teste.java | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
```

3) Fechar a janela da classe Java Teste no Eclipse e abrir novamente. Observar que as alterações realizadas remotamente foram aplicadas ao projeto Java.



git status

4) Reportar o estado do repositório.

$ git status
On branch main
Your branch is up to date with 'origin/main'.

```
PS C:\Users\DOCUMENTOS\Desktop\aula-git-github> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

nothing to commit, working tree clean

Resumo

- Comandos Básicos

- **git init** //Inicia um repositório e começa a observar as alterações do projeto. **git add .** //Realiza adições na staging area.
git commit -m 'mensagem' //Cria uma versão.
git remote add origin <url do repositório remoto> //Associar o repositório local ao repositório remoto.
git push origin main //Realizar envio do repositório local para o repositório remoto.

- Comandos Principais

- **git --version** //Reportar a versão do sistema.
git config //Configurar o sistema.
git init //Inicia um repositório e começa a observar as alterações do projeto.
git status //Reportar o estado do repositório.
git add . //Realiza adições na staging area.
git commit -m 'mensagem' //Cria uma versão.
git log //Histórico dos commits realizados.
git shortlog //Histórico resumido dos commits realizados.
git show <id do commit> //Apresenta informação sobre um commit.
git diff //Exibe diferenças entre alterações.
git remote add origin <url do repositório remoto> //Associar o repositório local ao repositório remoto.
git remote //Listar a associação criada.
git remote –v //Listar detalhes sobre a associação criada.
git push origin main //Realizar envio do repositório local para o repositório remoto.

git push -u origin main //Realizar envio do repositório local para o repositório remoto. **-u**: para não ter que informar “origin” e “main” nas próximas vezes que o repositório local tiver que ser enviado para o repositório remoto.
origin: repositório remoto.
main: para enviar do “branch” “main” local, para o “branch” “main” remoto. Se o “branch” “main” remoto não existir, ele é criado. Equivale a:
git push -u origin main:main
Que significa:
git push -u origin <branch local>:<branch remoto>
git push //Enviar do repositório local para o repositório remoto.

git fetch origin //Buscar o repositório remoto para o repositório local, sem realizar o “merge” dos “branchs”.

git pull //Realiza o “merge” dos “branchs”.
git pull origin //Buscar o repositório remoto para o repositório local, realizando o “merge” dos “branchs”.