



**PECE** Programa de  
Educação Continuada

Escola Politécnica da USP

# Introdução a Redes Neurais

Marlon Sproesser Mathias

# Versionamento

- O que é versionamento?
- Por que fazer controle de versões?
- Como fazer?

# Git

- Controle de versão
- Criado por Linus Torvalds em 2005 para ajudar no desenvolvimento do Linux
- Serve para projetos individuais ou colaborativos



# Git

Por que usar?

- Fácil de testar novas funções
- Permite desenvolvimento compartilhado
- Mantém histórico de mudanças



# Instalando



## No Linux (Ubuntu)

```
sudo apt install git
```



## Windows

<https://git-scm.com/download/win>

```
winget install --id Git.Git -e --source winget
```

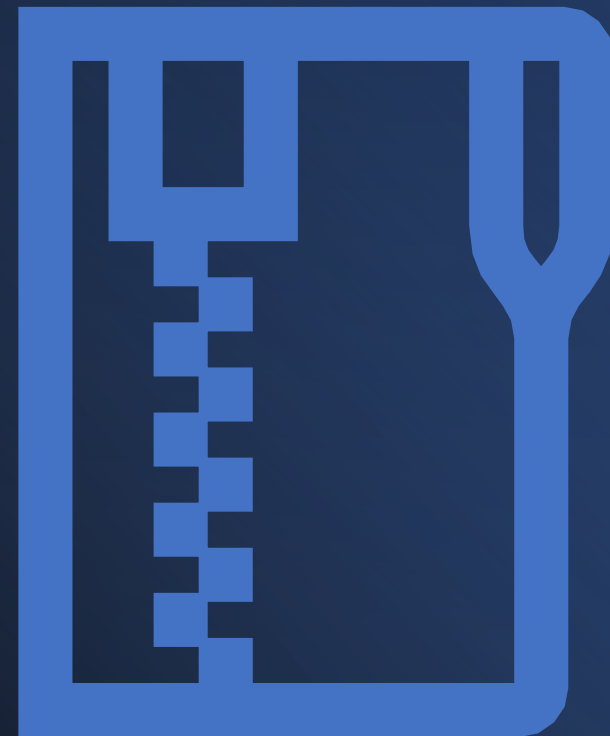
# Configurando o git

## Configurar nome e email

- `git config --global user.name "NomeDeUsuario"`
- `git config --global user.email email@email.com`
- `git config --local user.name "NomeDeUsuario"`
- `git config --local user.email "email@email.com"`

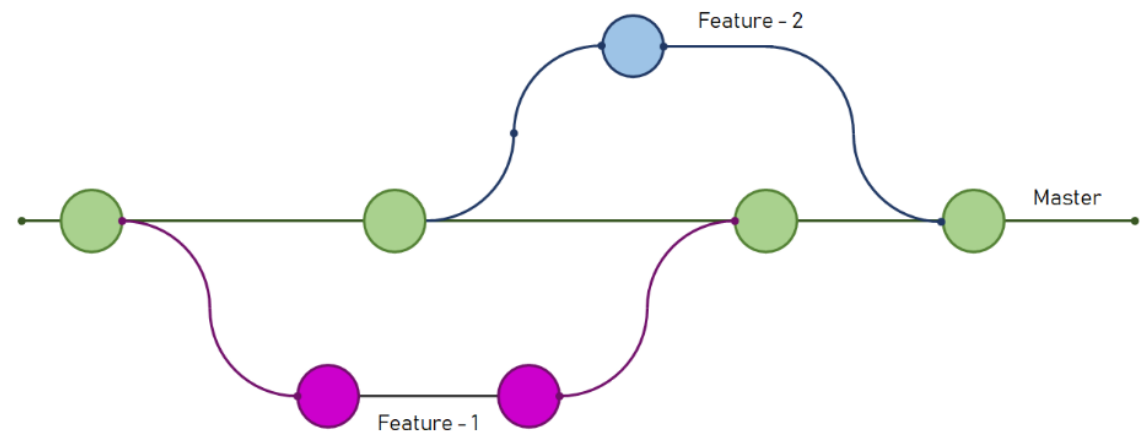
# Repository

- É o projeto
- Pasta com todo o código
- Contém uma pasta .git
- git init



# Branch

- “Galho”
- Uma linha de desenvolvimento dentro do projeto
- `git branch [nome]`
- `git checkout`
- `git merge`

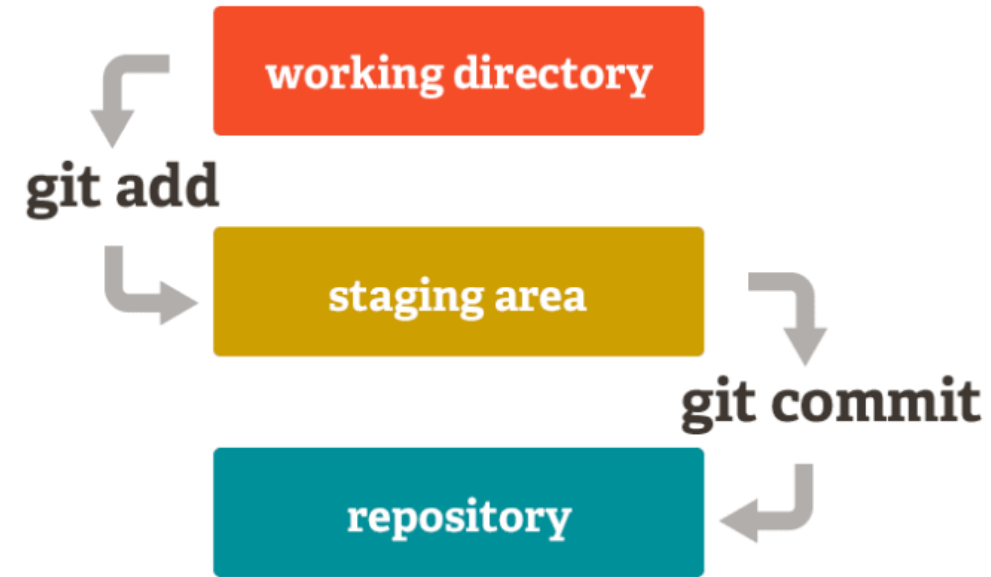


<https://digitalvarys.com/git-branch-and-its-operations/>



# Staging area

- `git add [arquivo]`
- `git commit`
- `git status`
- `git diff`
- O git trabalha com arquivos em 3 níveis
  - Working directory
  - Staging area
  - Repository



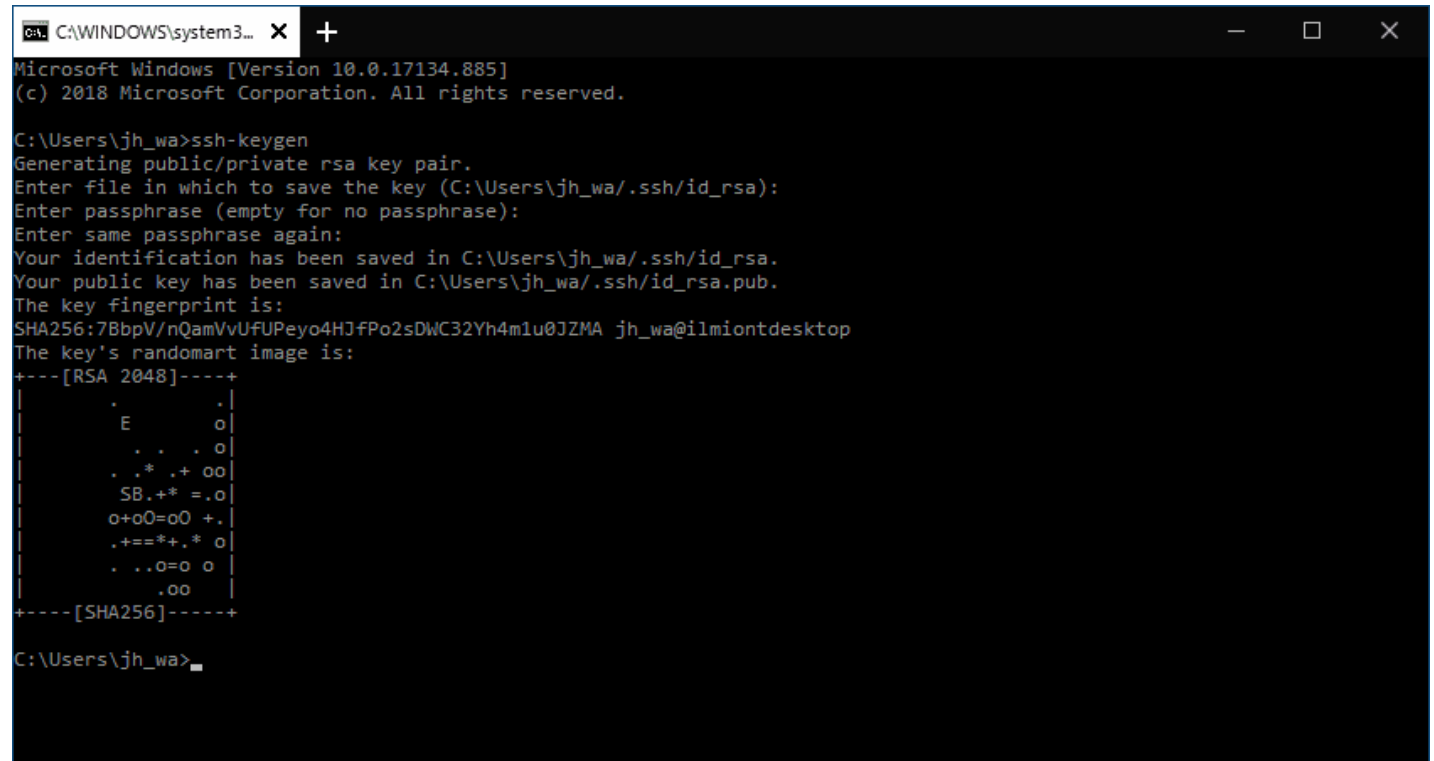
# Como compartilhar git?

- GitHub, GitLab, BitBucket
- São plataformas que hospedam repositórios
- Projetos podem ser privados ou públicos



# Como integrar com o GitHub

- Criar uma chave pública
- ssh-keygen



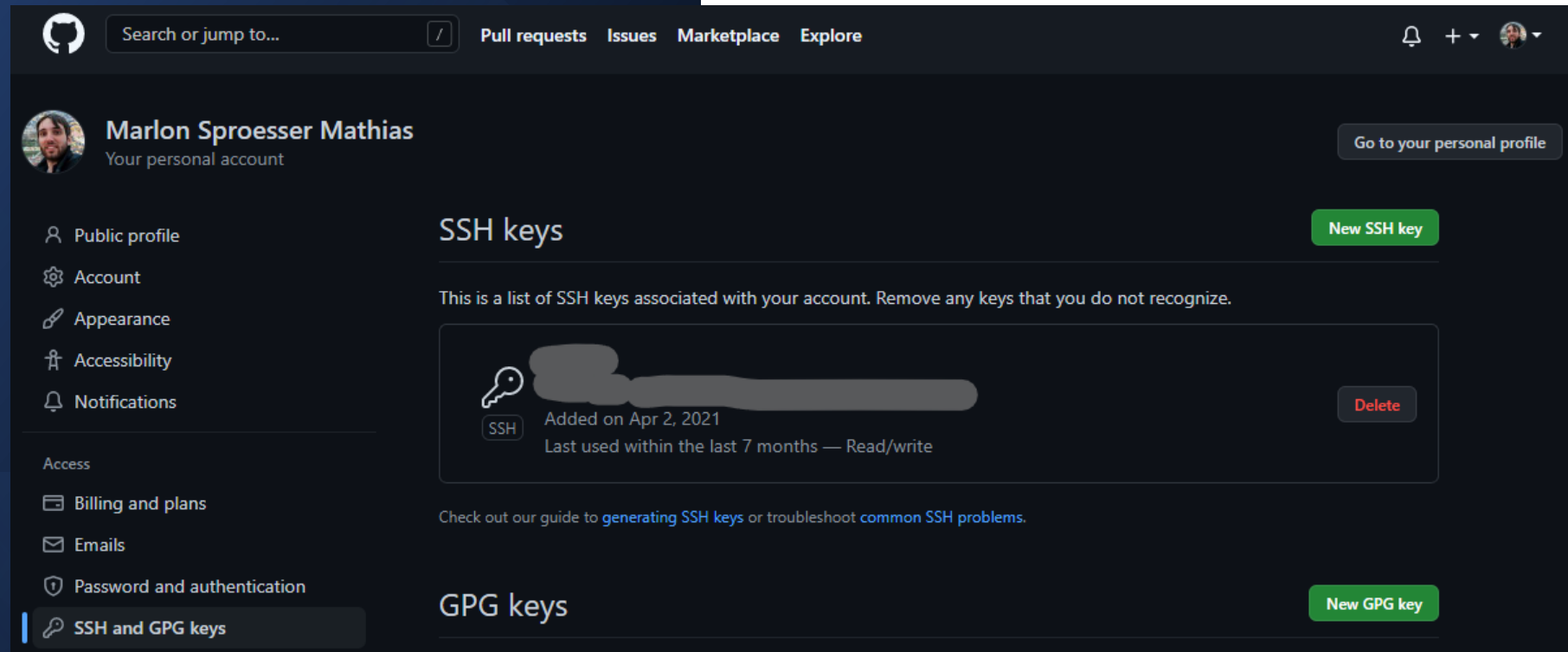
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\jh_wa>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\jh_wa/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\jh_wa/.ssh/id_rsa.
Your public key has been saved in C:\Users\jh_wa/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7BbpV/nQamVvUfUPeyo4HJfPo2sDWC32Yh4m1u0JZMA jh_wa@ilmiontdesktop
The key's randomart image is:
+----[RSA 2048]-----+
|          .          |
|      E            o|
|      . . . o       |
|    . * . + oo      |
|   SB . + * = . o   |
|  o+oO=oO +.       |
| .+==*+. * o       |
| . ..o=o o         |
|      .oo          |
+----[SHA256]-----+

C:\Users\jh_wa>
```

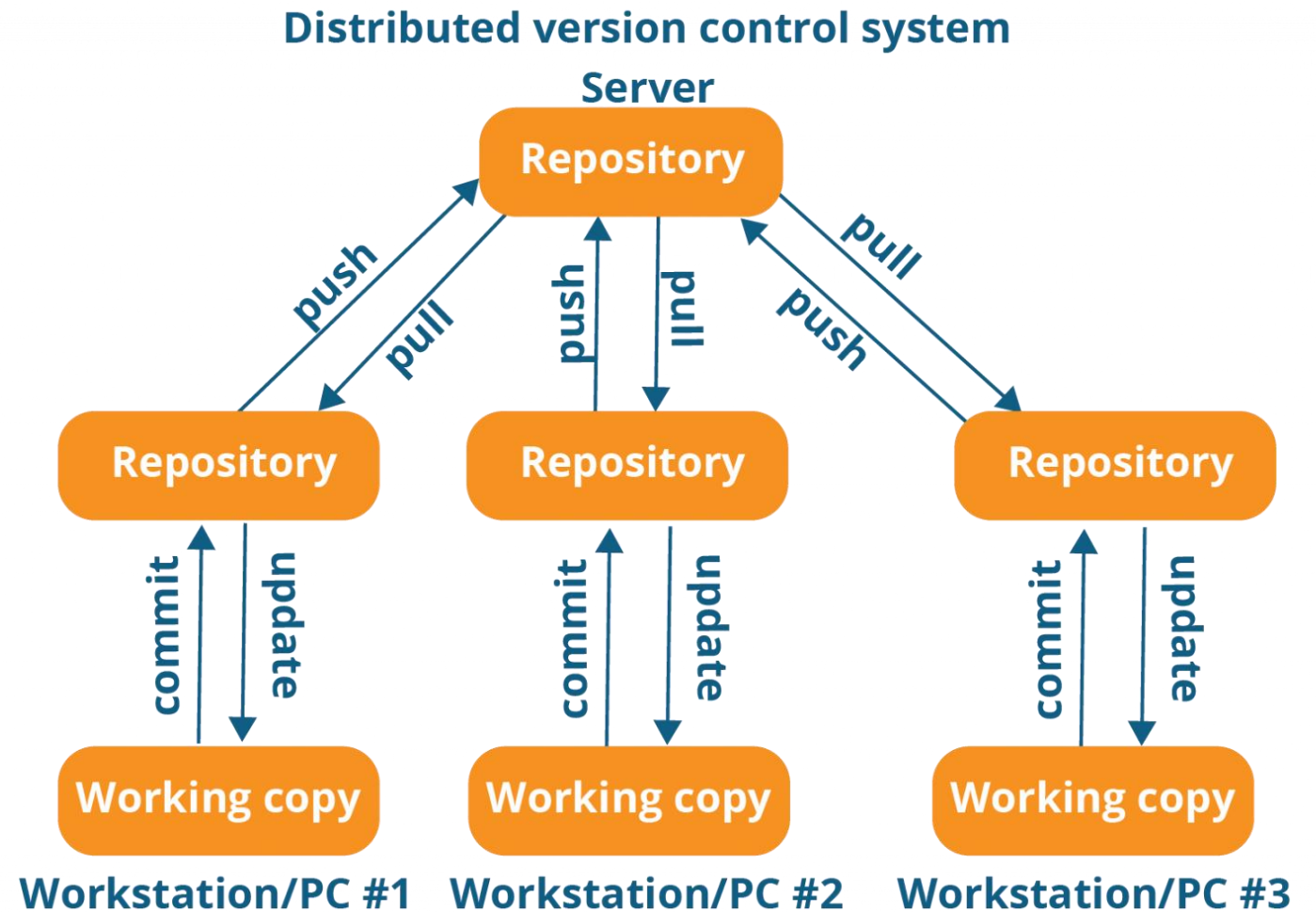
<https://www.onmsft.com/how-to/how-to-generate-an-ssh-key-in-windows-10>

# Como integrar com o GitHub



Adicionar chave ao GitHub  
(<https://github.com/settings/keys>)

# Servidor remoto (GitHub)



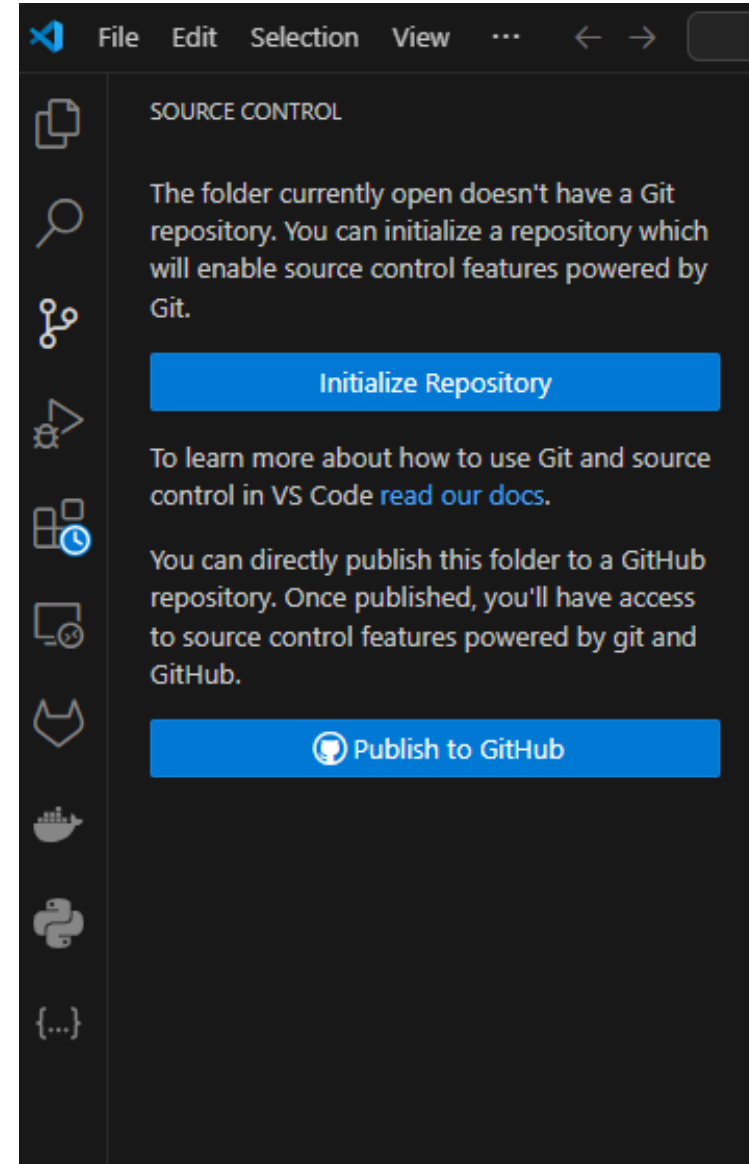
[https://www.dotnetwisdom.com/Products/Git\\_Basic\\_Commands](https://www.dotnetwisdom.com/Products/Git_Basic_Commands)

# Terminologia

- git
- GitHub, GitLab, BitBucket
- Repository
- Branch
- Commit
- Stash
- Push/Pull
- Fork/Clone

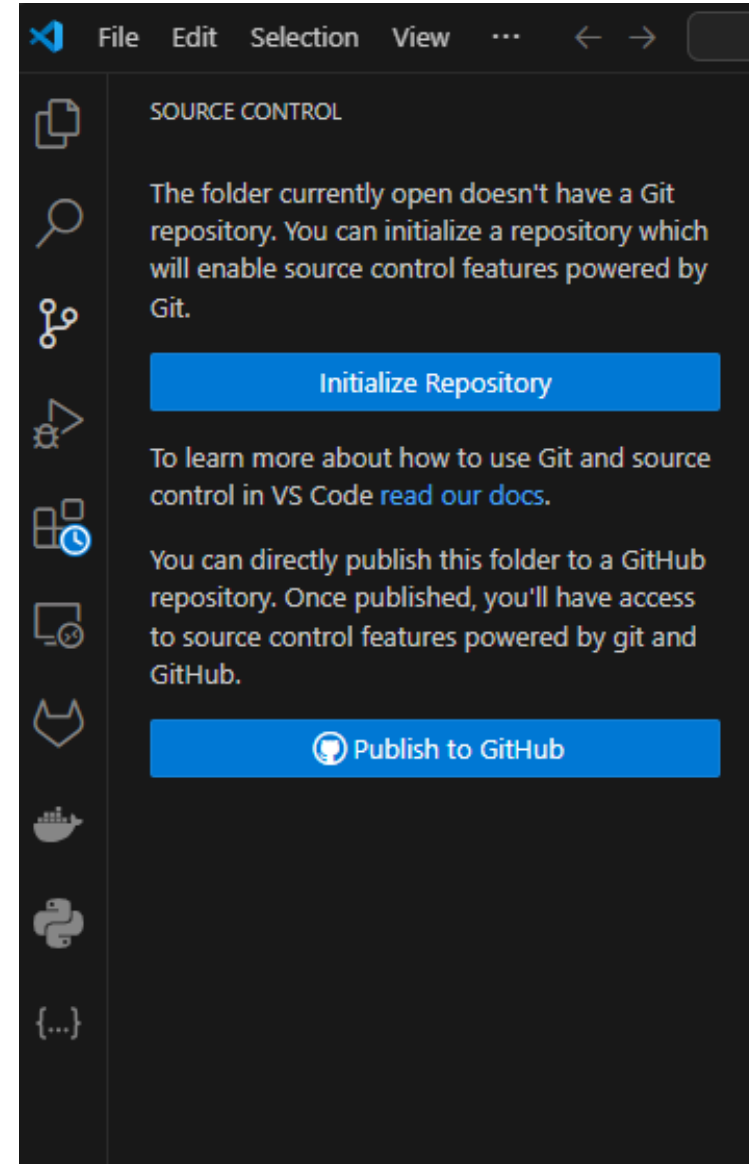
# Exemplo

- Vamos usar o VSCode nesse exemplo
- As operações serão feitas pela interface do VSCode
- Existem comandos equivalentes que podem ser dados no terminal



# Inicializando o repositório

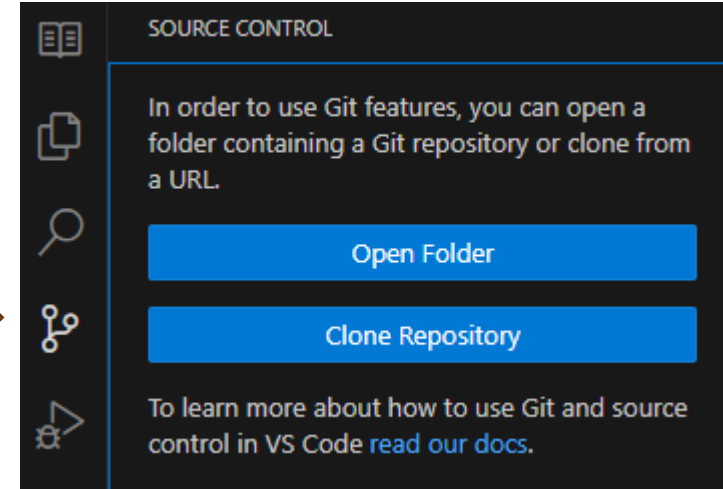
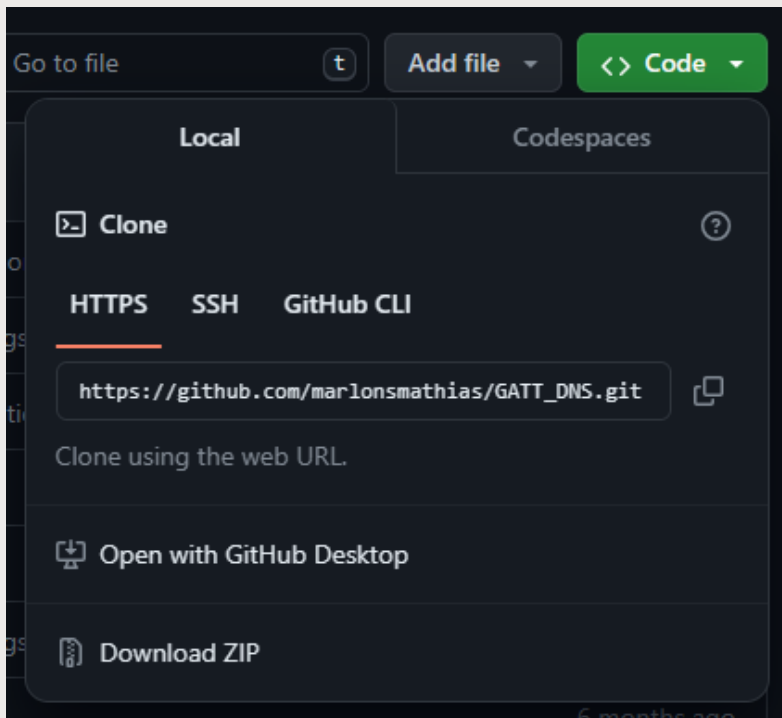
- Cria a pasta .git
- git init





# Inicializando o repositório

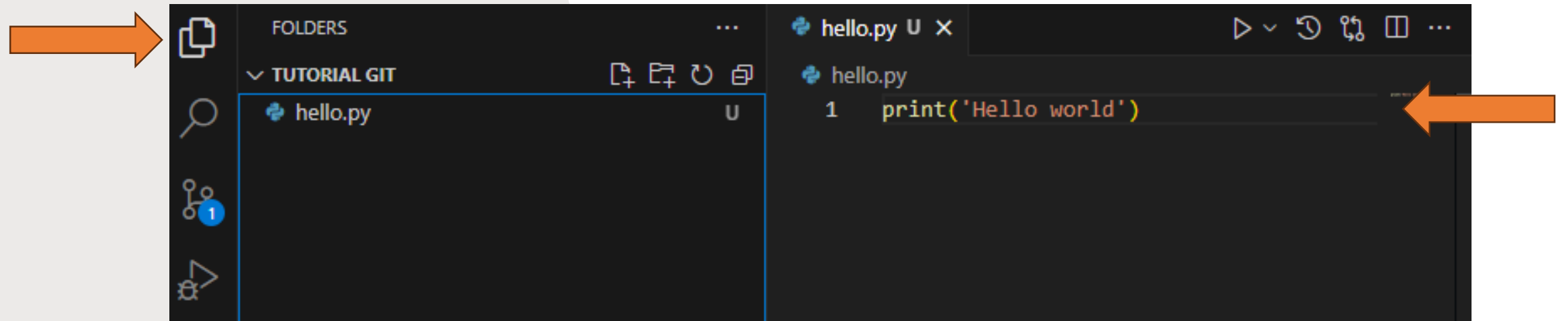
- Uma alternativa é clonar um repositório já existente
- `git clone [url] [destino]`



# Criando os arquivos

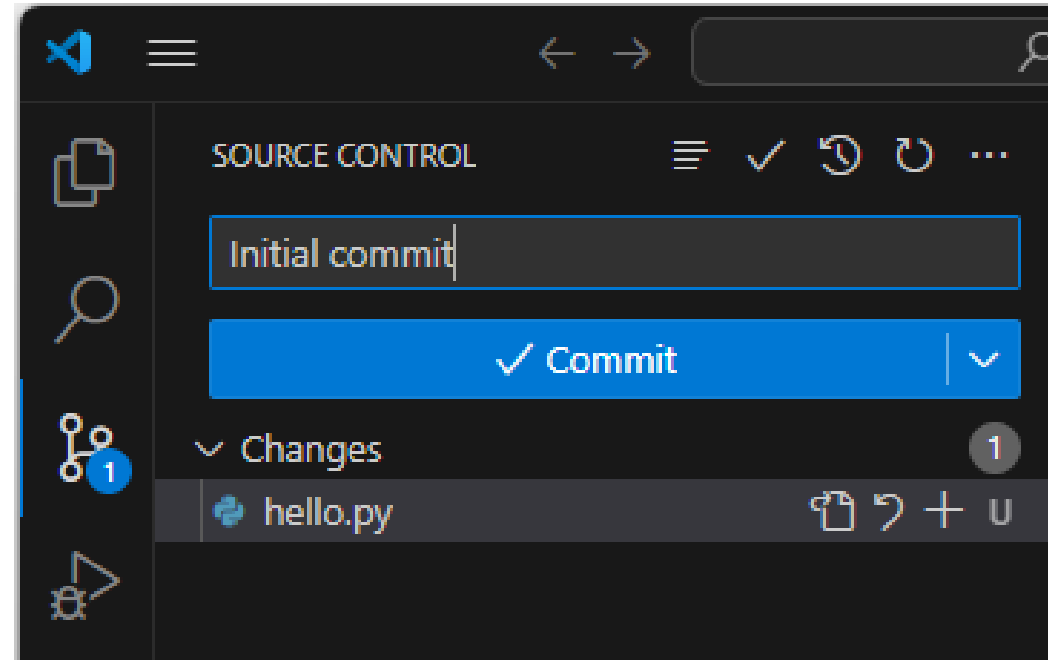
- Arquivo hello.py

```
print('Hello world')
```



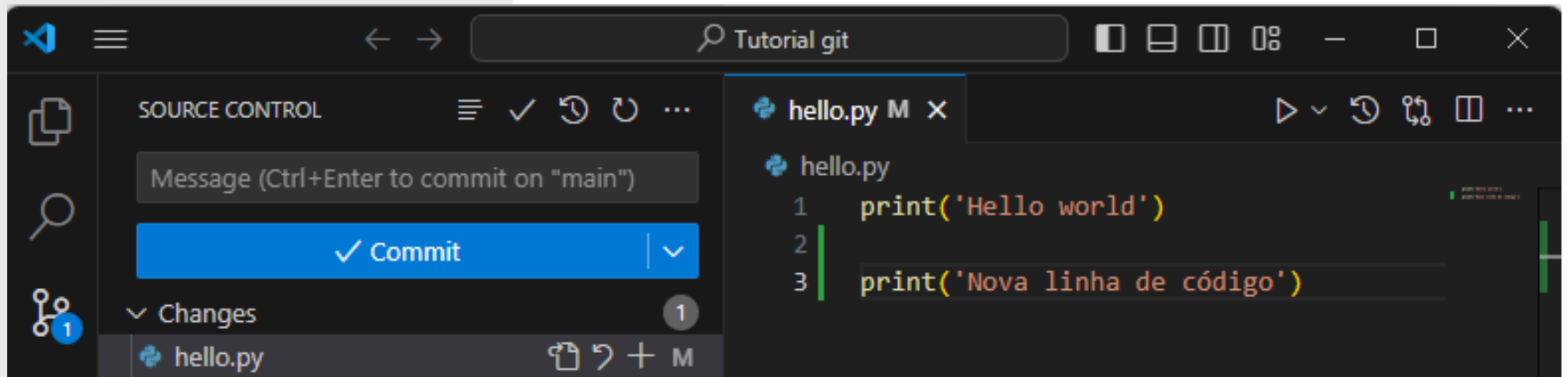
# Fazendo o primeiro commit

- Voltando à aba do git
  - As mudanças são identificadas
  - Devemos nomear o commit
- 
- `git add .`
  - `git commit -m "Initial commit"`



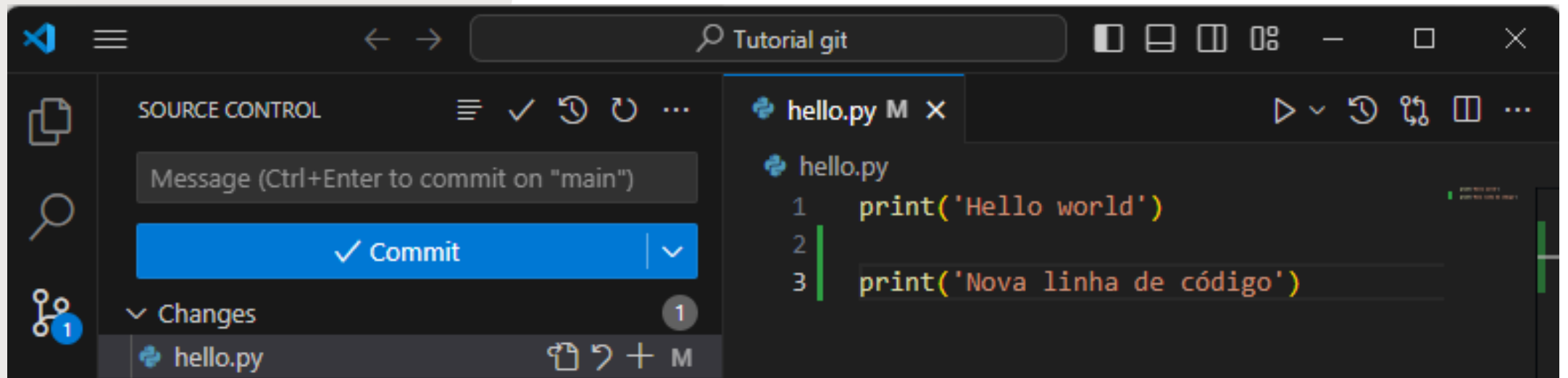
# Alterando o código

- Se alterarmos o código original, o git identifica as novas linhas
- As linhas alteradas são destacadas no editor



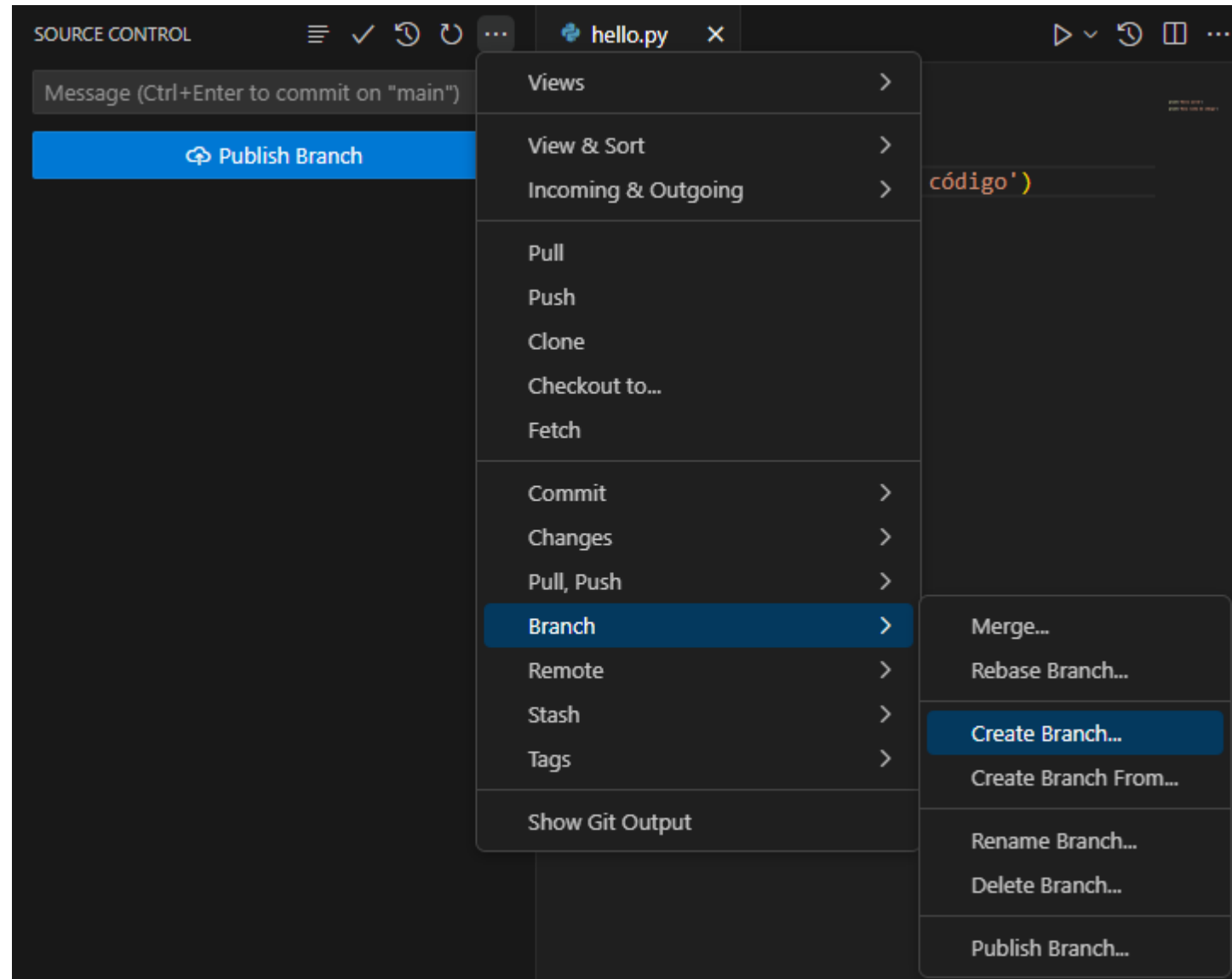
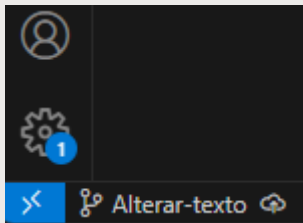
# Novo commit

- Vamos chamar essa alteração de “Nova linha” e fazer um commit



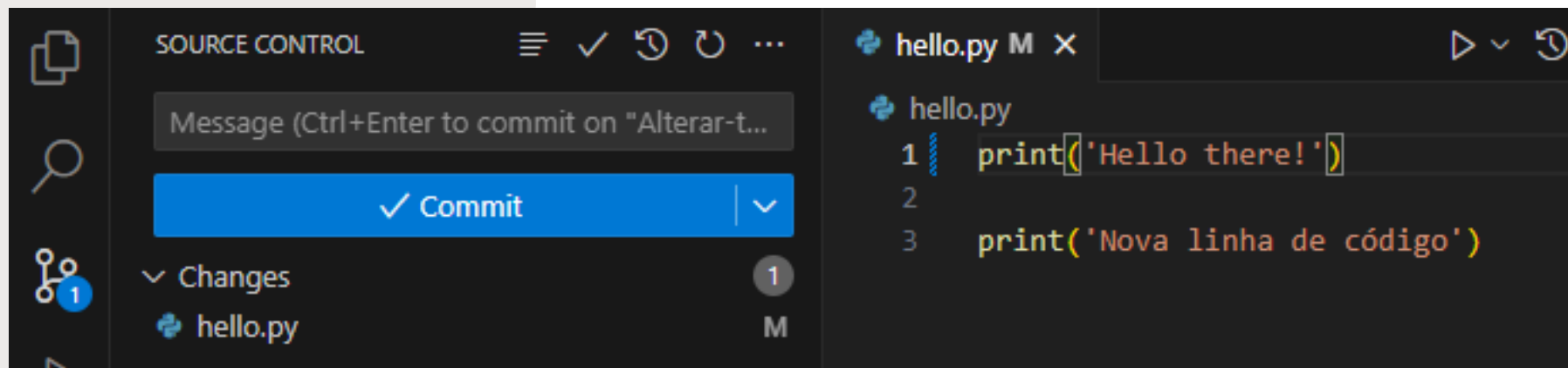
# Fazendo um branch

- Vamos criar um novo branch chamado “Alterar-texto”
- git branch Alterar-texto



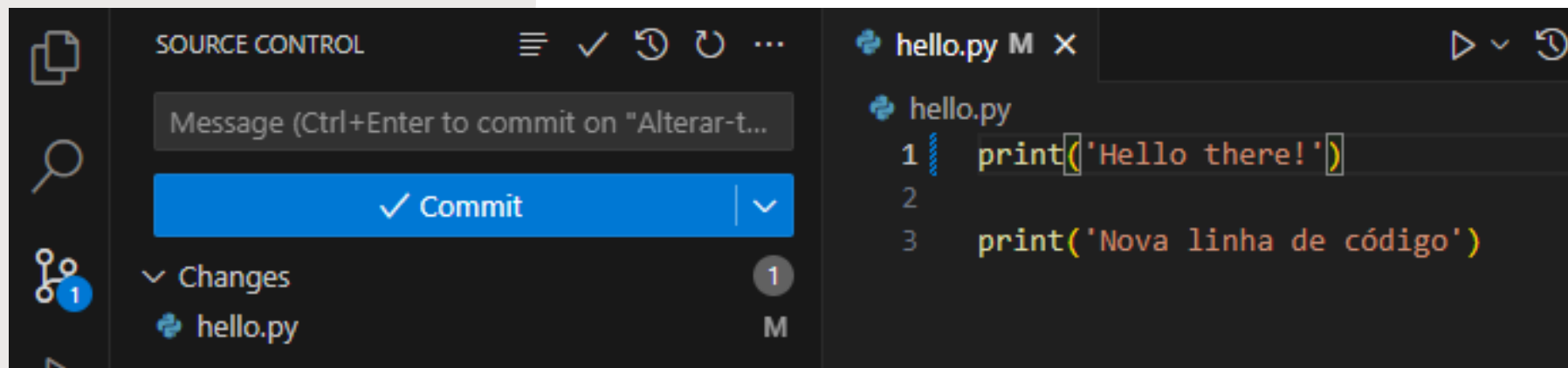
# Mudanças no código

- Vamos criar um novo branch chamado “Alterar-texto”
- `git branch Alterar-texto`



# Novo commit

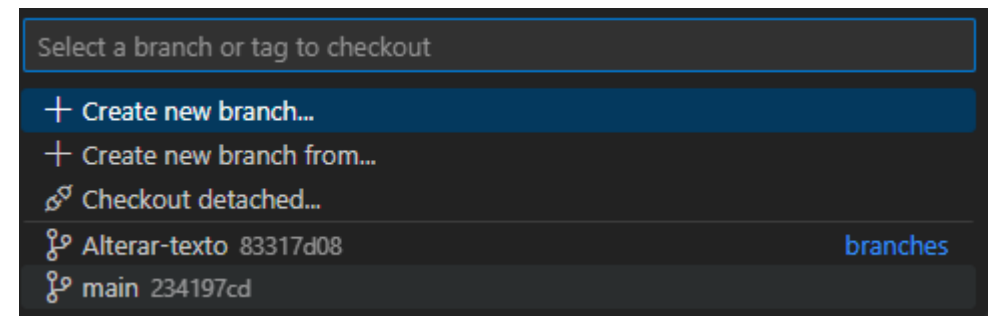
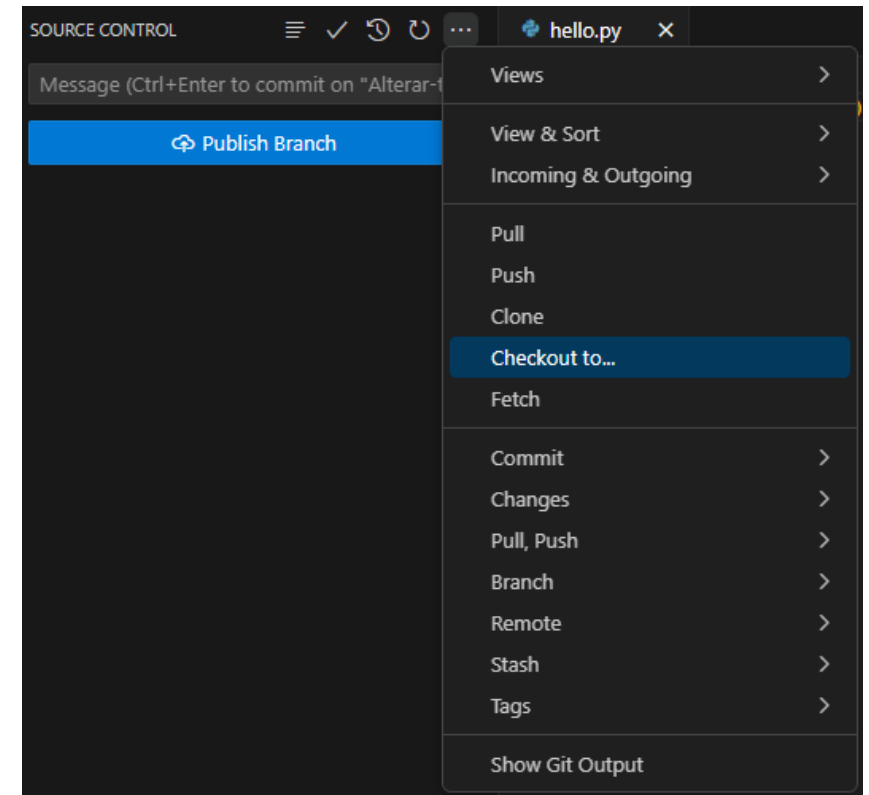
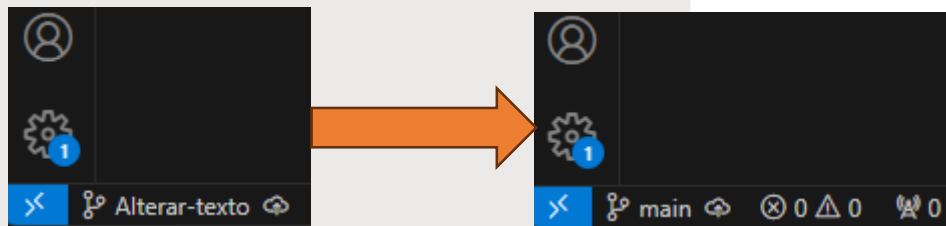
- Vamos fazer um commit nesse branch, chamado de “texto alterado”





# Alternar branch

- Podemos alternar entre branches
- Atenção: Mudanças que não fazem parte de um commit serão perdidas
- `git checkout [branch]`



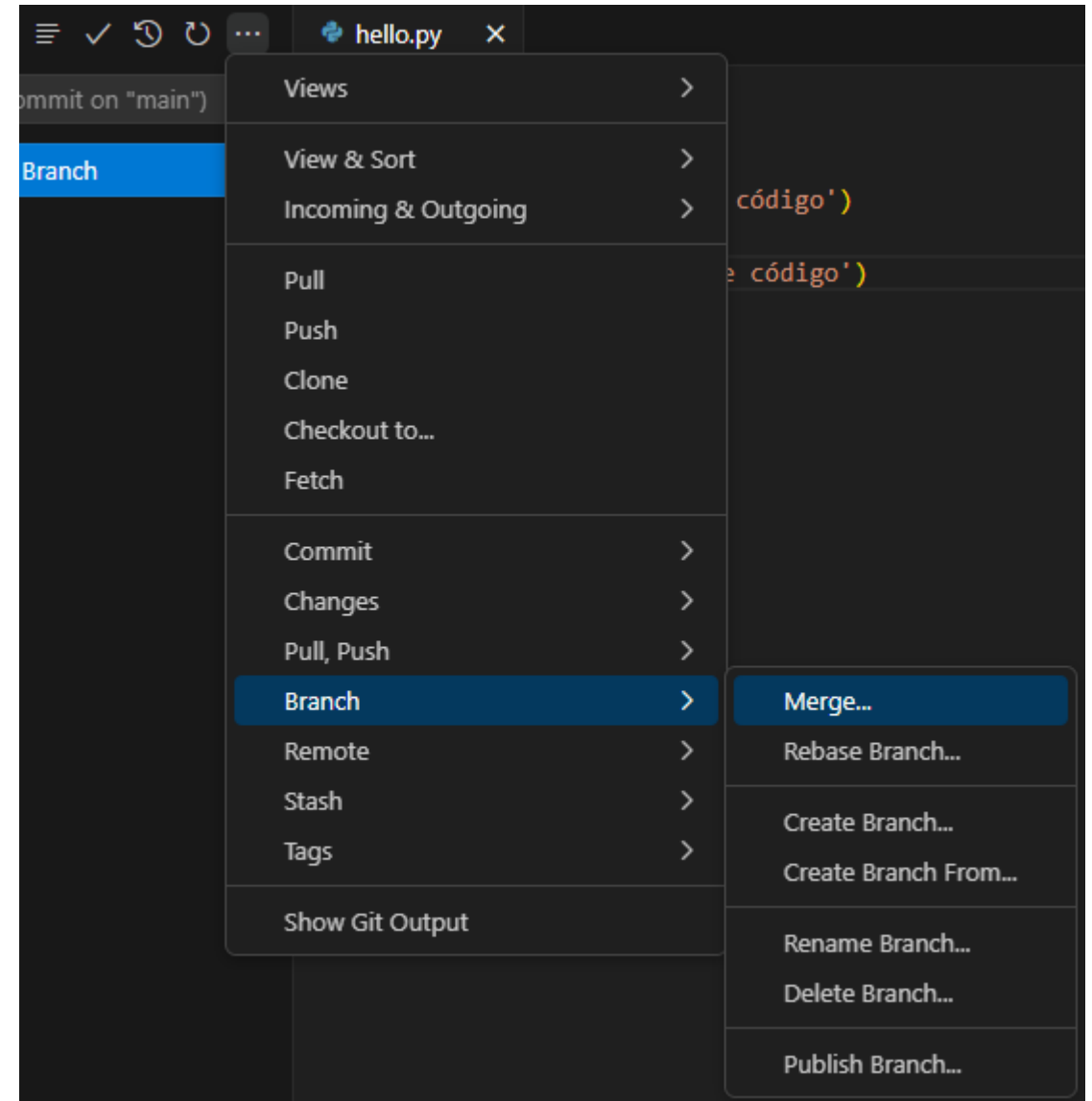
# Mais mudanças no main

- Podemos continuar alterando o main
- commit “Outra linha”

```
hello.py M X
hello.py
1  print('Hello world')
2
3  print('Nova linha de código')
4
5  print('Outra linha de código')
```




# Mesclando as mudanças

- Vamos trazer as mudanças do outro branch para o principal



# Entendendo o que aconteceu

- Existem diversas extensões e programas para visualizar as alterações com o git
- Por exemplo, git graph:

Graph		Description	Date	Author	Commit
	 <b>main</b>	Merge branch 'Alterar...	13 Mar 202...	Marlon Spr...	0ccde859
		Outra linha	13 Mar 202...	Marlon Spr...	b8331e5a
	 <b>Alterar-texto</b>	Texto alterado	13 Mar 202...	Marlon Spr...	83317d08
		Nova linha	13 Mar 202...	Marlon Spr...	234197cd
		Initial commit	13 Mar 202...	Marlon Spr...	2abe9fb0

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

[xkcd.com/1296](https://xkcd.com/1296)