



REVISAO

GIT E GITHUB



OIIIIII!



Eu sou a May(hhara Morais)!

Sou engenheira de software backend, ex-aluna reprograma e, atualmente, trabalho com Python no Grupo Boticário.

Sou de Pernambuco, tenho 5 gatinhos lindos/lindas (faltou 1 na foto 😭) e vou conduzir a aula de revisão hoje.

AGENDA

- **Kahoot!** - Bora relembrar uns conceitos?
- **Git & GitHub**
 - Uso e necessidade.
 - Comandos básicos: fetch, status, add, commit, push.
- **Fluxo de Trabalho**
 - Fork e clone do repositório da Reprograma.
 - Alterações locais e Pull Request.
- **VS Code & Terminal**
- **Markdown**
 - Formatar texto: negrito, itálico, emojis.

Lembretes

- Está com dúvida?
 - pergunte POR FAVORRRR!
- Não tenham medo de errar!
- Bebam água.



@mayhharamorais



KAHOOT

link.aqui.com

@mayharamorais



GITE
GITHUB

GIT E GITHUB

Ferramenta de linha
de comando para
controle de versão.

Serviço online para
hospedar e
compartilhar
repositórios Git.

GIT E GITHUB

Conceitos Importantes

- Repositório Remoto x Local:** Diferenças e interações.
- Branches:** Importância e gerenciamento.
- Merge vs. Rebase:** Estratégias para integrar mudanças.

Github

- Plataforma de Colaboração**
- Repositórios:** Armazenamento do projeto e código.
- Forks:** Experimentar mudanças independentemente.
- Pull Requests:** Propor e revisar mudanças no código.

Ferramenta de linha de comando para controle de versão.

Serviço online para hospedar e compartilhar repositórios Git.

Comandos Básicos do Git

- init:** Inicializar um novo repositório Git.
- status:** Verificar o estado do repositório.
- add:** Adicionar arquivos ao próximo commit.
- commit:** Registrar mudanças no repositório.
- push:** Enviar alterações para o repositório remoto.
- pull:** Atualizar o repositório local com as mudanças do remoto.
- merge:** Combinar alterações de diferentes branches.
- fetch:** Baixar conteúdo de um repositório remoto.
- branch:** Gerenciar branches.

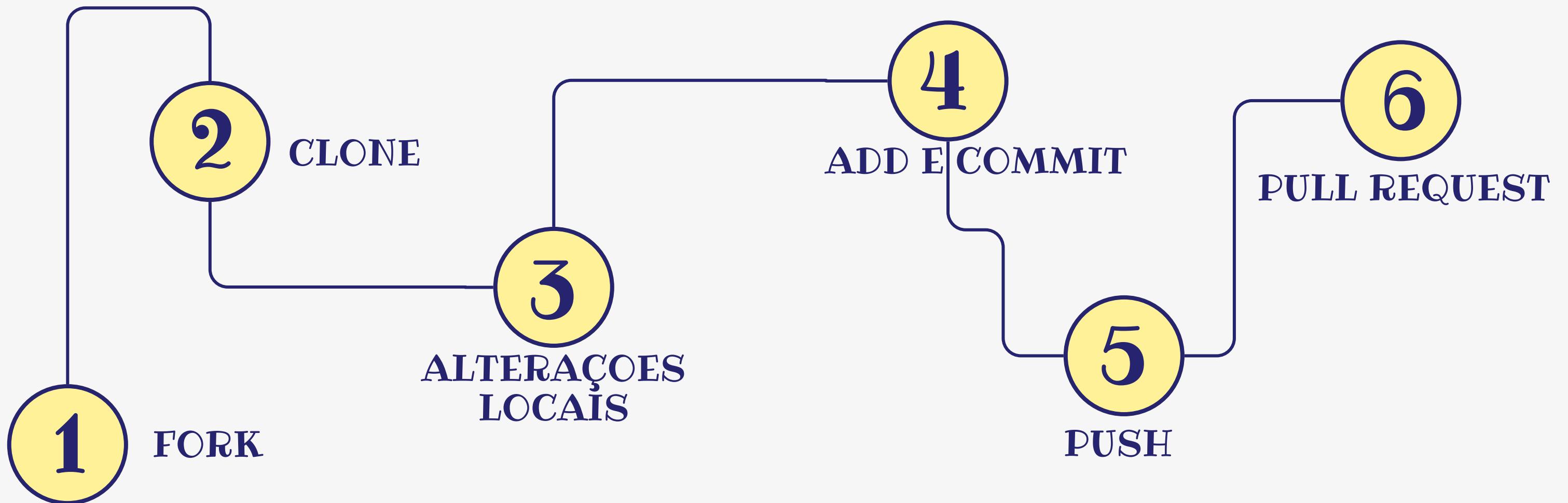
Ferramentas Auxiliares

- GitHub Desktop:** Interface gráfica para operações Git.
- Git GUIs:** Interfaces gráficas como Sourcetree ou GitKraken.
- Git Bash:** Terminal Git para usuários Windows.
- Integração VSCode com o Git:** Usar Git diretamente do editor de código.

FLUXO DE TRABALHO

Processo padrão para
desenvolver e
atualizar sistemas ou
programas.

FLUXO DE TRABALHO



No repositório que você quer forkar, clique em Fork

1- Fork
Crie uma cópia do repositório da reprograma para sua conta.

Create a new fork
copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Repository name *
ian / on29-python-s03-logica-II
 on29-python-s03-logica-II is available.

Description (optional)
On29

Branch
the **main** branch only
Switch back to reprograma/on29-python-s03-logica-II by adding your own branch. [Learn more.](#)

Create fork

Voce será transferida pra essa tela, onde voce pode, por exemplo, alterar o nome do repositório forkado, colocar descrição, etc.
Clique em "Create fork".

on29-python-s03-logica-II Public
forked from reprograma/on29-python-s03-logica-II

main 1 Branch 0 Tags
This branch is up to date with reprograma/on29-python-s03-logica-II:main

Dariellysantos fix
assets Initial commit
exercicios fix
material Initial commit
README.md feat: script da aula

Code ▾
Local Codespaces
Clone
HTTPS SSH GitHub CLI
git@github.com:mflilian/on29-python-s03-logica-II...
Use a password-protected SSH key.
Open with GitHub Desktop
Download ZIP

Python On29
Readme Activity 0 stars 0 watching 0 forks
Releases No releases published Create a new release
Packages No packages published Publish your first package

Turma Online On29 | Python | Semana 3 | 2024 | Darielly Santos Python Package using Anaconda

1- Fork

Crie uma cópia do repositório da reprograma para sua conta.

2- Clone

Baixe o repositório "forkado" para sua máquina local.

```
[N094711] as mayhhara.morais in ~/Documents/may 09:49:01
└─> git clone git@github.com:mflilian/on29-python-s03-logica-II.git
Cloning into 'on29-python-s03-logica-II'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 29 (delta 6), reused 26 (delta 5), pack-reused 0
Receiving objects: 100% (29/29) 39.53 KiB | 4.39 MiB/s, done.
Resolving deltas: 100% (6/6), done.

[N094711] as mayhhara.morais in ~/Documents/may 09:49:41
└─>
```

Depois de forkar, ele aparece como um novo repositório seu.
Para clonar, clique no botão "Code" e copie o endereço do repositório (HTTPS ou SSH)

Abra o terminal no seu computador dentro da pasta que você quer que ele fique. Aqui, no caso, eu clonei dentro de uma pasta chamada "may".

Se retornou 100% done no final da mensagem, o seu clone foi sucesso!

The screenshot shows a GitHub repository named "ON29-PYTHON-S03-LOGICA-II" opened in Visual Studio Code. The repository contains several files and folders related to Python logic exercises.

Left Panel (VS Code Explorer):

- OPEN EDITORS:** Shows "maior_idade.py" and "README.md".
- ON29-PYTHON-S03-LOGICA-II:** Contains "assets", "exercicios", and "para-casa" folders.
- para-casa folder contents:** "atividade_dia_a_dia.txt", "erros.txt", "exercicio_farmacia.py", and "instrucoes-pull-request.md".

Code Editor (maior_idade.py):

```

1 # - Explicação
2
3 idade = int(input("Qual é sua idade? "))
4 carteira = input("Você tem dinheiro na carteira? ")
5
6 if idade >= 18:
7     print("Você é maior de idade!")
8 elif idade >= 12:
9     print("Você é menor de idade, mas maior que 12 anos!")
10 else:
11     print("Você é menor de idade!")
12

```

Right Panel (README.md):

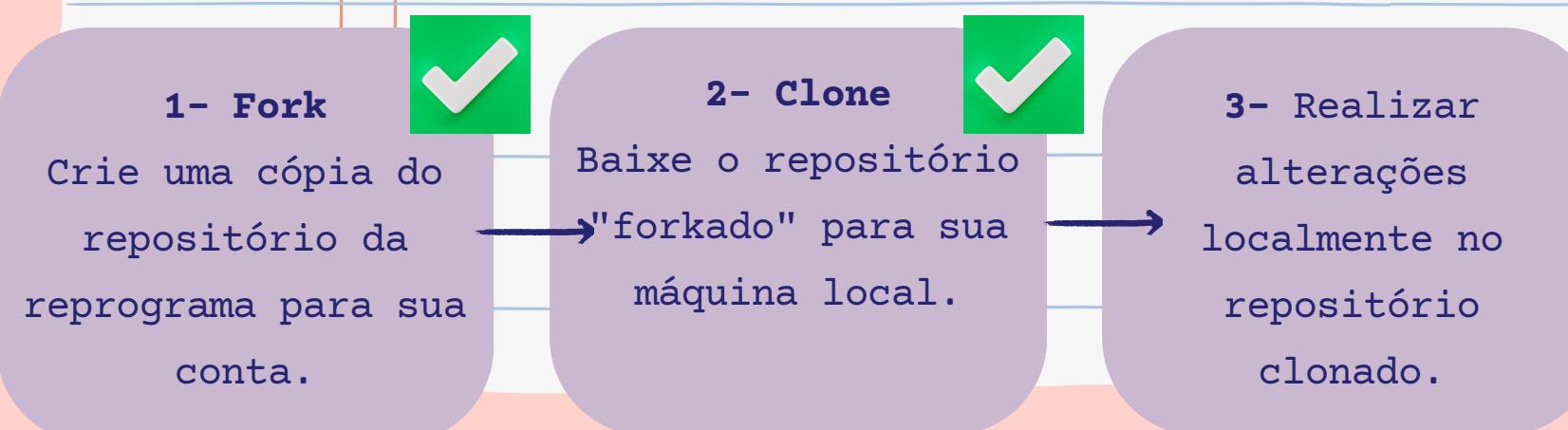
```

1 README.md > # Condicionais, operadores lógicos, erros exceções
2 # Condicionais, operadores lógicos, erros exceções
3 *** Material da aula
4
5 *** Exercícios
6 * [Exercício para sala](https://github.com/mflilian/repo)
7 * [Exercício para casa](https://github.com/mflilian/repo)
8
9 *** Links Úteis
10 - [Python If ... Else](https://www.w3schools.com/python/python_conditions.asp)
11 - [Built-in Exceptions](https://docs.python.org/3/library/exceptions.html)
12 - [Debugging -VScode](https://code.visualstudio.com/docs/editor/troubleshooting)
13 - [Python Try Except](https://www.w3schools.com/python/python_try_except.asp)
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

Annotations:

- "alteração 1" is highlighted in the code editor around line 11.
- "alteração 2" is highlighted in the code editor around line 12.



```
mayhhara.morais in ~/Documents/may/on29-python-s03-logica-II on (main)*
(./0.0) 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:   README.md
    modified:   exercicios/para-casa/maior_idade.py

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

No terminal, rodei o comando `git status` para ver o estado atual do meu repositório. Note que ele me informa que existem dois arquivos modificados: README.md e maior_idade.py

```
mayhhara.morais in ~/Documents/may/on29-python-s03-logica-II on (main)*
(./0.0) git add .
```

[N094711] as mayhhara.morais in ~/Documents/may/on29-python-s03-logica-II on (main)*
(./0.0) git commit -m "alterei o readme; adicionei a resposta do exercício xpto"

Vamos **adicionar** essas mudanças e depois **commitá-las**. Ou seja, primeiro, usamos o comando **git add** para preparar os arquivos para o commit. E, em seguida, fazemos o commit das mudanças, salvando as alterações com uma descrição/mensagem.

1- Fork

Crie uma cópia do repositório da reprograma para sua conta.



2- Clone

Baixe o repositório "forkado" para sua máquina local.



Realizar alterações localmente no repositório clonado.



4- Add e Commit

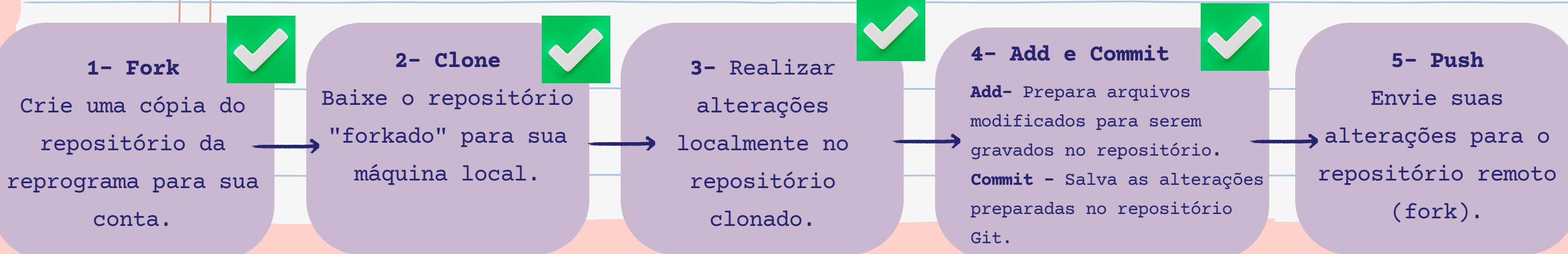
Add - Prepara arquivos modificados para serem gravados no repositório.
Commit - Salva as alterações preparadas no repositório Git.

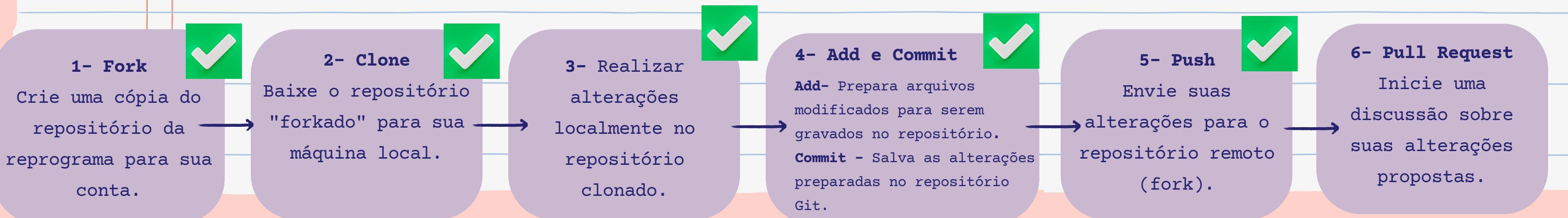
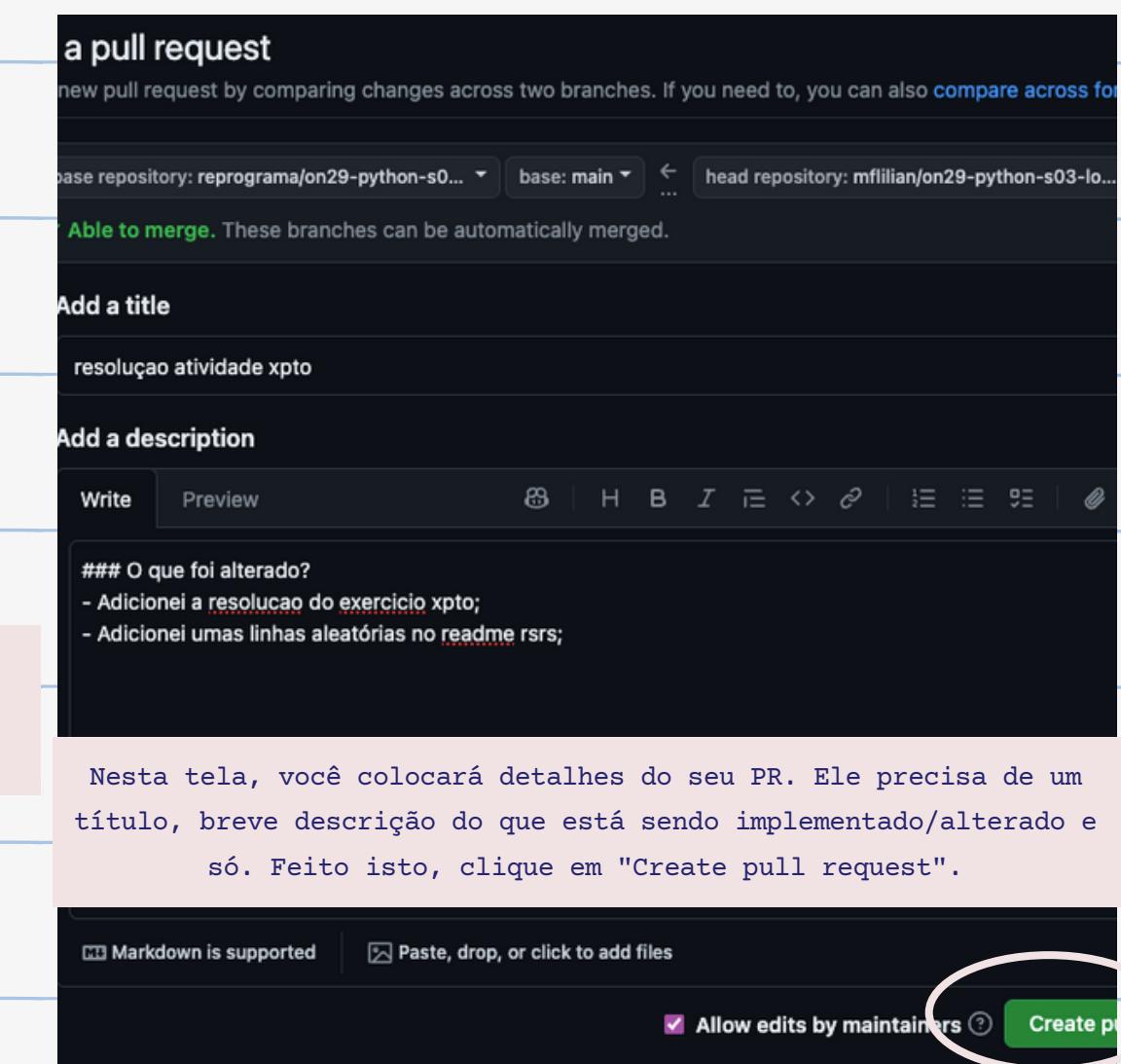
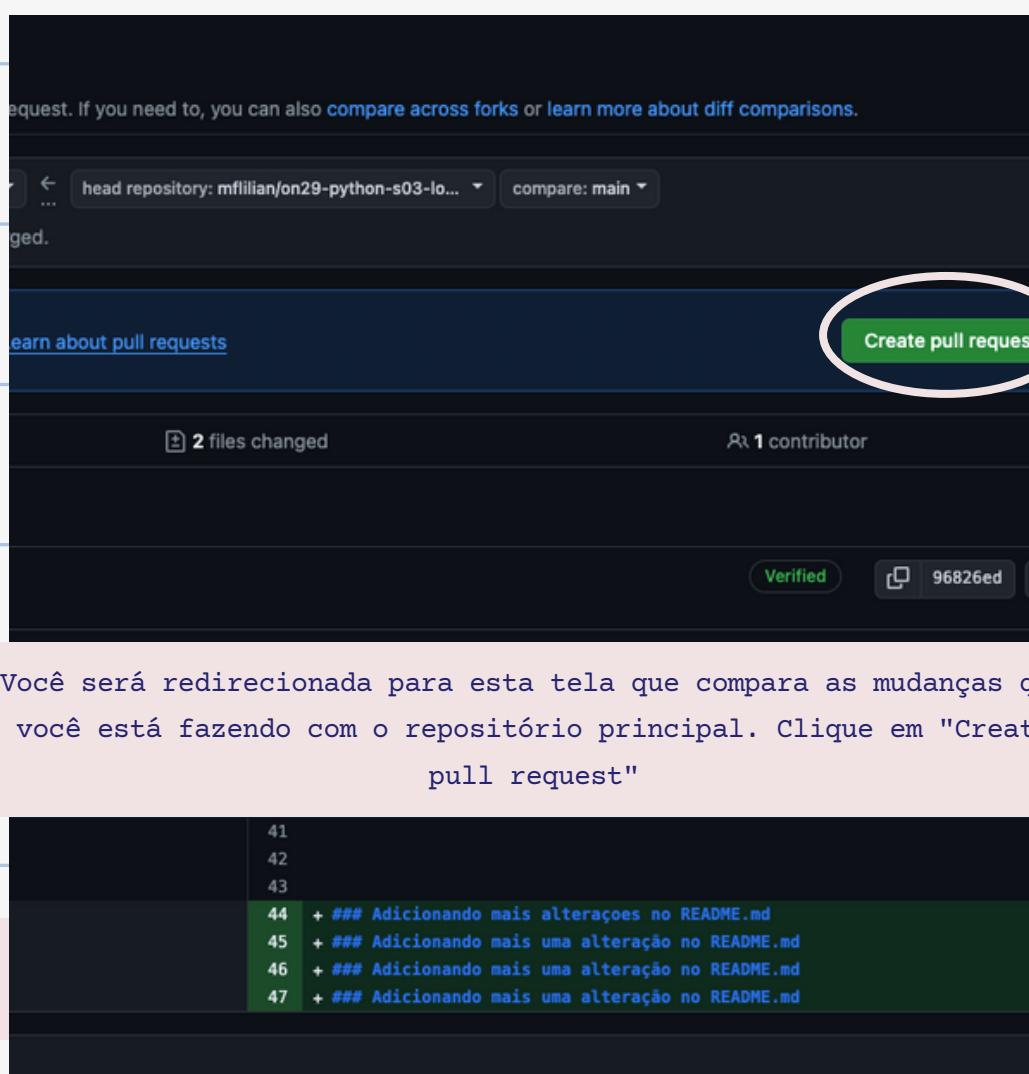
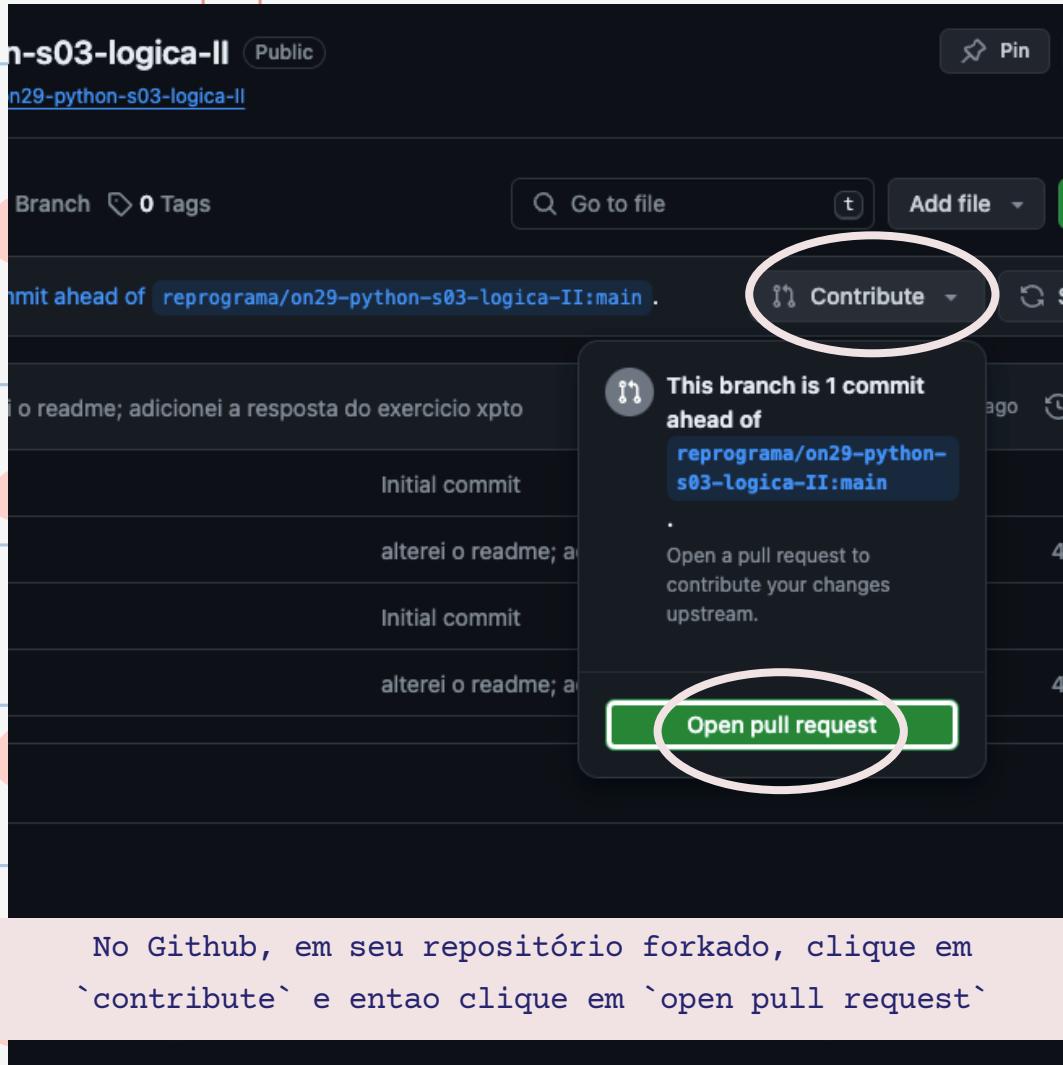
```
[ mayhhara.morais in ~/Documents/may/on29-python-s03-logica-II on (main)*
└(./0*)/ git add .

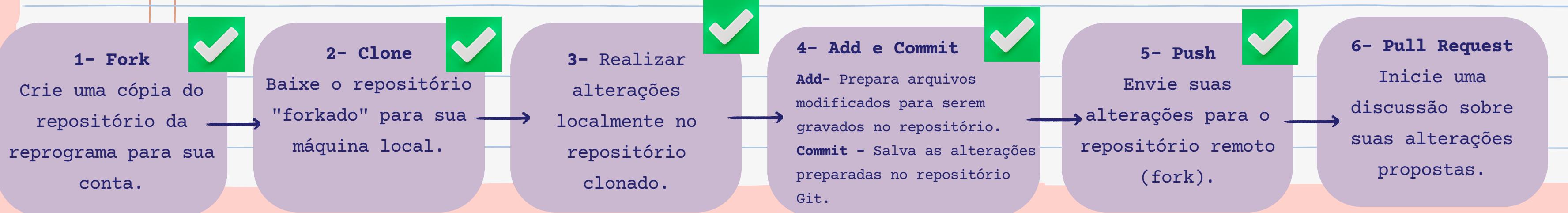
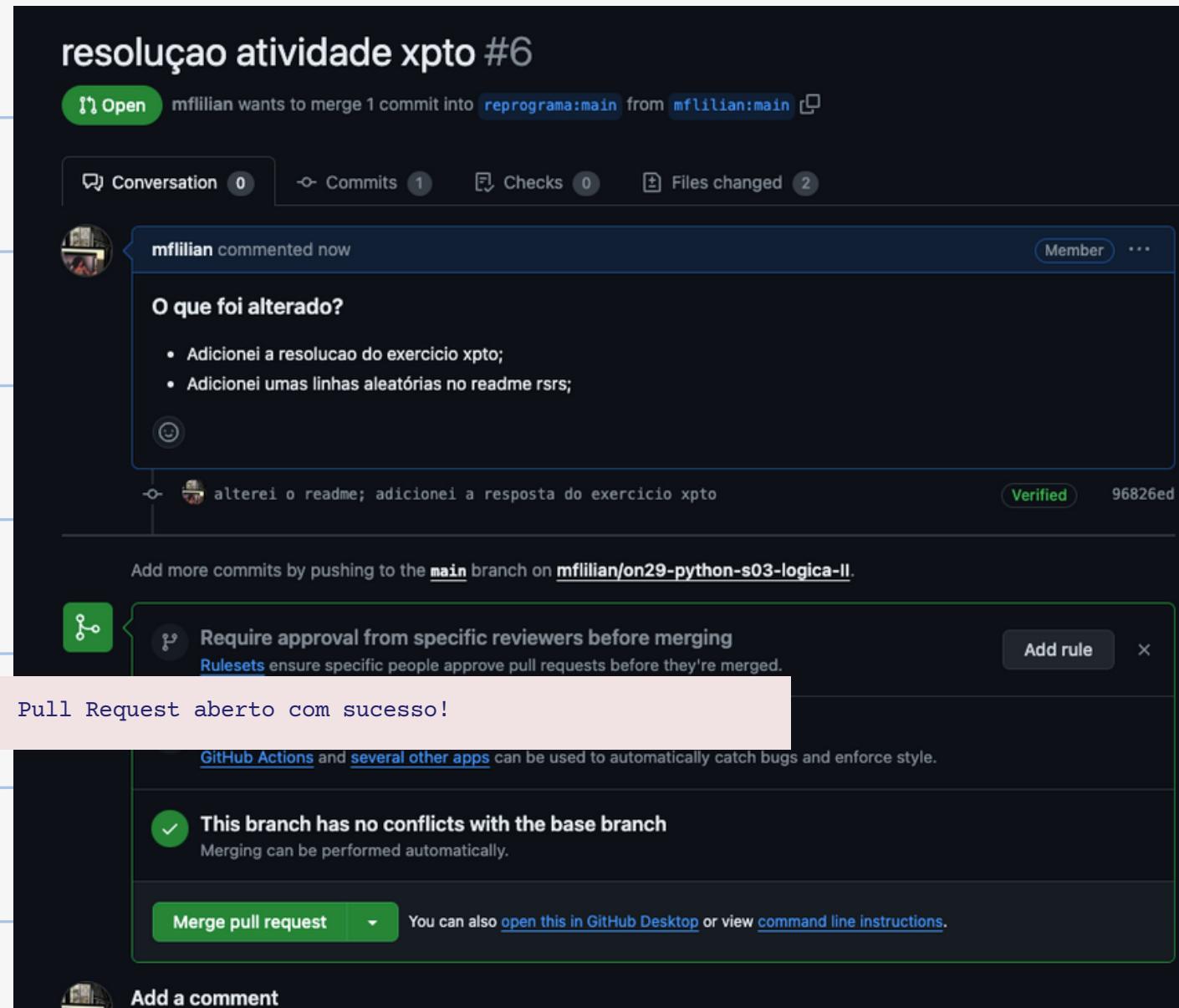
[N094711] as mayhhara.morais in ~/Documents/may/on29-python-s03-logica-II on (main)*
└(./0*)/ git commit -m "alterei o readme; adicionei a resposta do exercício xpto"
[main 96826ed] alterei o readme; adicionei a resposta do exercício xpto
 2 files changed, 16 insertions(+)

[N094711] as mayhhara.morais in ~/Documents/may/on29-python-s03-logica-II on (main)
└(./0*)/ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.40 KiB | 1.40 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:mflilian/on29-python-s03-logica-II.git
```

Agora, ainda no terminal, vamos enviar as alterações para o repositório forkado através do comando `git push`.

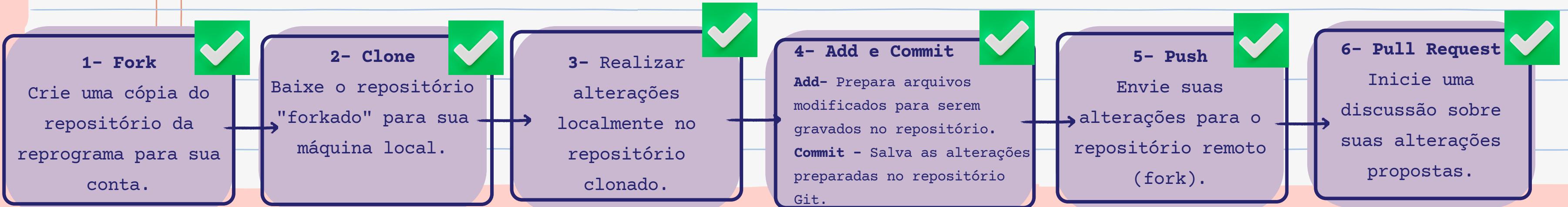






FLUXO DE TRABALHO

Processo padrão para desenvolver e atualizar sistemas ou programas.



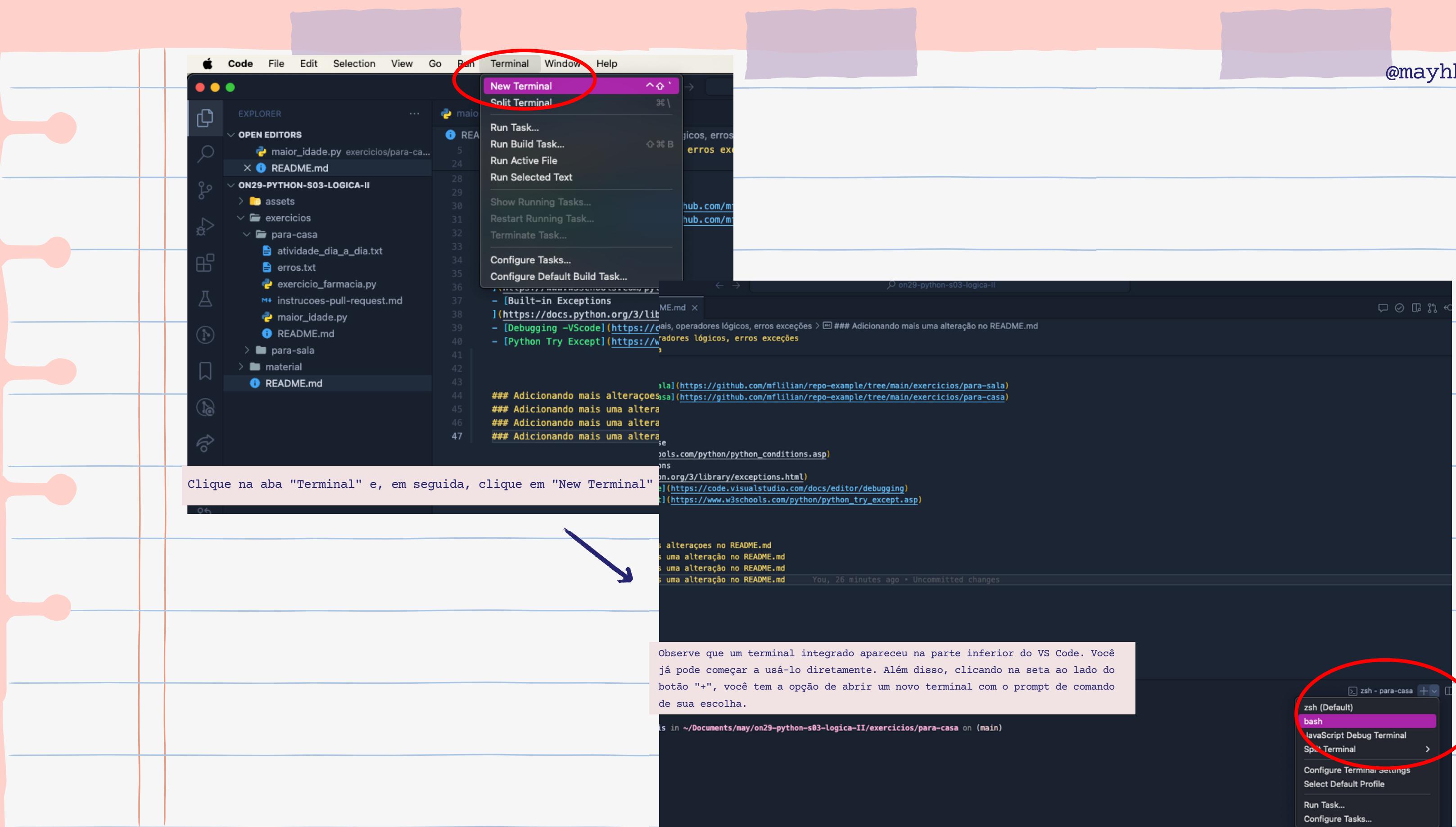
VS CODE E TERMINAL

Executa comandos sem
sair do editor

Codificação e terminal
na mesma interface.

Configurações adaptáveis
para diferentes
necessidades.

Reduz a troca de
contexto e otimiza o
workflow.





Linguagem de marcação usada para
formatar texto de maneira simples e
intuitiva, ideal para web.

MARKDOWN

ELEMENTOS DE MARKDOWN



FORMATAR TEXTO

Exemplo:

Negrito: ****texto****

Itálico: **texto**

EMOJIS

Exemplo: `:emoji_name:`

IMAGENS

Exemplo: ![alt text](url)

GIFS

Exemplo: ![alt text](url do gif)

OBRIGADA!

13

