

## **MATERIAL COMPLEMENTAR 1 - ROTEIRO DA ATIVIDADE DE EXPERIMENTAÇÃO**

Este material foi desenhado para servir como um guia prático para os participantes e, ao mesmo tempo, como um instrumento de coleta de dados para a pesquisa, permitindo observar as barreiras e os avanços conceituais no primeiro contato com o pensamento computacional.

### **Roteiro da Atividade: Da Inércia à Interação**

#### **1. Objetivos Pedagógicos**

- Familiarizar os licenciandos com um ambiente de desenvolvimento profissional (Visual Studio Code).
- Introduzir os conceitos e a prática de controle de versão com Git e GitHub.
- Articular conceitos matemáticos básicos com sua implementação na linguagem de programação Python.
- Fomentar a colaboração, a depuração de erros e a autonomia na resolução de problemas

#### **2. Estrutura da Aula**

- Apresentação da importância do VS Code e do GitHub
- Configurações de instalação
- Realizações práticas de edição de código e armazenamento

#### **3. Projeto Prático**

- Criando e Publicando um Repositório
- Criar um novo projeto no VS Code.
- Inicializar um repositório Git local.
- Fazer o primeiro commit e publicar no GitHub
- Clonar um repositório existente e fazer alterações.

## 4. Recursos Online

- [Visual Studio Code Para Iniciantes - Resumo e Dicas!](#)
- [Introdução ao Git e GitHub para Iniciantes](#)

## 5. Momento 1: Configuração do Ambiente e o Primeiro Código

**Foco:** *Superar a barreira inicial de configuração e executar o primeiro programa, estabelecendo uma base de confiança e familiaridade com as ferramentas.*

### 5.1. O que é o Visual Studio Code (VS Code)?

O Visual Studio Code (VS Code) é um editor de código-fonte gratuito e de código aberto desenvolvido pela Microsoft. Ele é amplamente utilizado por desenvolvedores devido à sua flexibilidade, desempenho e uma vasta gama de funcionalidades. Algumas características importantes do VS Code são:

- a) **Multiplataforma:** o VS Code está disponível para Windows, macOS e Linux, permitindo que desenvolvedores trabalhem em diferentes sistemas operacionais com a mesma ferramenta.
- b) **Editor de Código Leve e Rápido:** Embora seja um editor poderoso, o VS Code é leve e rápido, oferecendo uma experiência de desenvolvimento ágil.
- c) **Suporte a Várias Linguagens de Programação:** suporta uma ampla variedade de linguagens de programação, incluindo Python, Javascript, Typescript, C++, Java, entre outras. Isso é possível graças ao seu sistema de extensões.
- d) **Extensões e Personalização:** através do Marketplace de Extensões, os usuários podem adicionar funcionalidades específicas ao VS Code, como suporte a novas linguagens, temas, ferramentas de depuração, entre outros. Isso permite uma personalização completa do ambiente de desenvolvimento.
- e) **Integração com Git e GitHub:** o VS Code possui integração nativa com o Git, facilitando o controle de versão diretamente no editor. Além disso, ele oferece suporte para trabalhar com repositórios do GitHub, tornando a colaboração e o gerenciamento de código mais simples.

**f) Terminal Integrado:** o editor inclui um terminal integrado, permitindo que os desenvolvedores executem comandos de linha de comando sem sair do ambiente de desenvolvimento.

**g) Depuração Avançada:** o editor oferece ferramentas de depuração avançadas, permitindo que os desenvolvedores configurem breakpoints, inspecionem variáveis e controlem a execução do código de maneira eficiente.

**h) IntelliSense:** é um recurso que fornece autocompletar inteligente, sugestões de código e documentação em tempo real, ajudando os desenvolvedores a escrever código de forma mais rápida e com menos erros.

Alguns benefícios do uso do VS Code são o aumento da produtividade proporcionado por suas funcionalidades e extensões, a vasta e ativa comunidade de usuários que oferece suporte e recursos, e as atualizações frequentes da Microsoft que garantem melhorias contínuas e novas funcionalidades. Em suma, o VS Code é apresentado como uma excelente opção para desenvolvedores de todos os níveis devido à sua simplicidade, poder e flexibilidade.

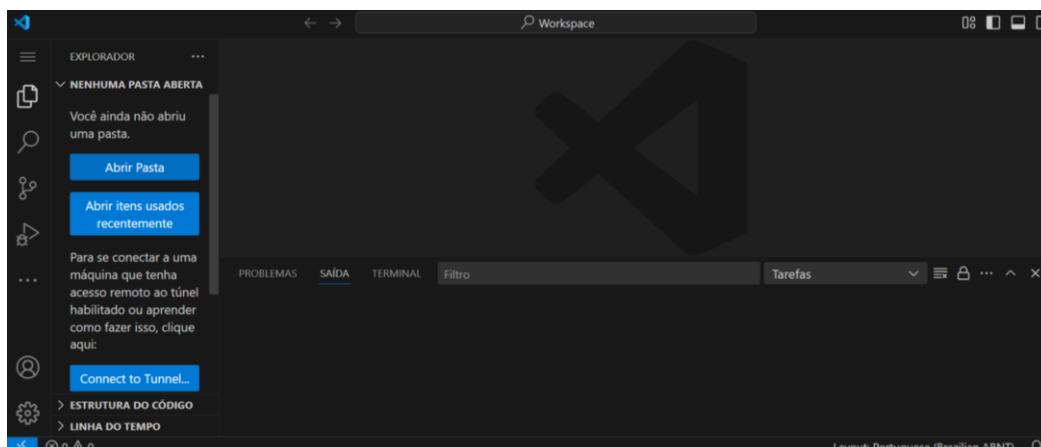
#### **Links que usaremos nas aulas:**

- **VS Code Online:** Você pode usar o Visual Studio Code diretamente no navegador através do [VS Code Online](#).
- **(opcional) Download do VS Code para PC:** Para baixar e instalar o VS Code no seu computador, acesse a página de [download do Visual Studio Code](#).

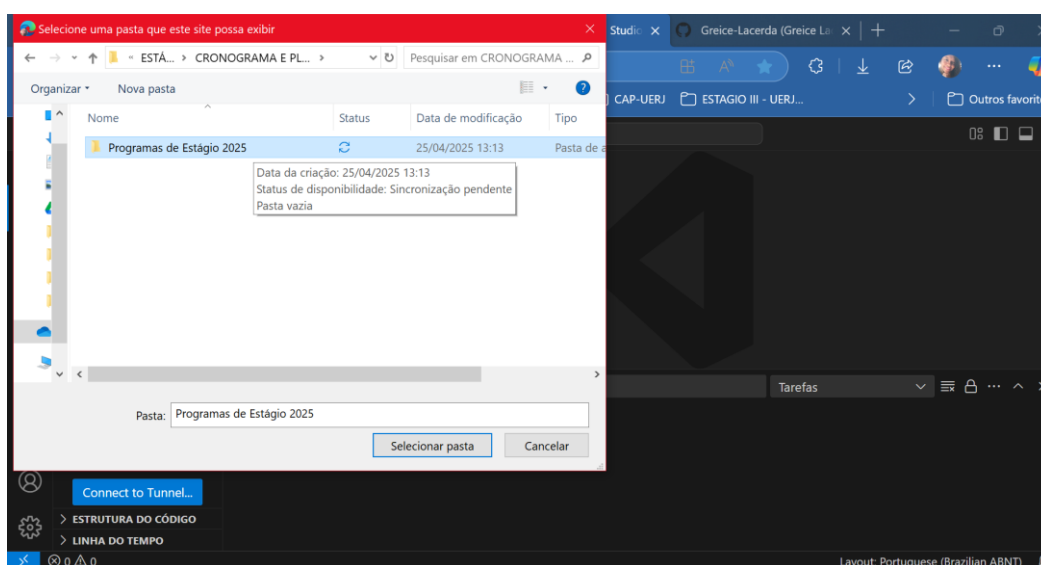
## **5.2. Primeiro programa no VS Code Online da aula de Estágio.**

1) Crie uma pasta na área de trabalho do computador e nomeie-a de “Programas da Aula de Estágio”. E dentro dessa pasta, crie outra pasta nomeada “.git” (**Essa pasta será importante para o salvamento posterior dos códigos**).

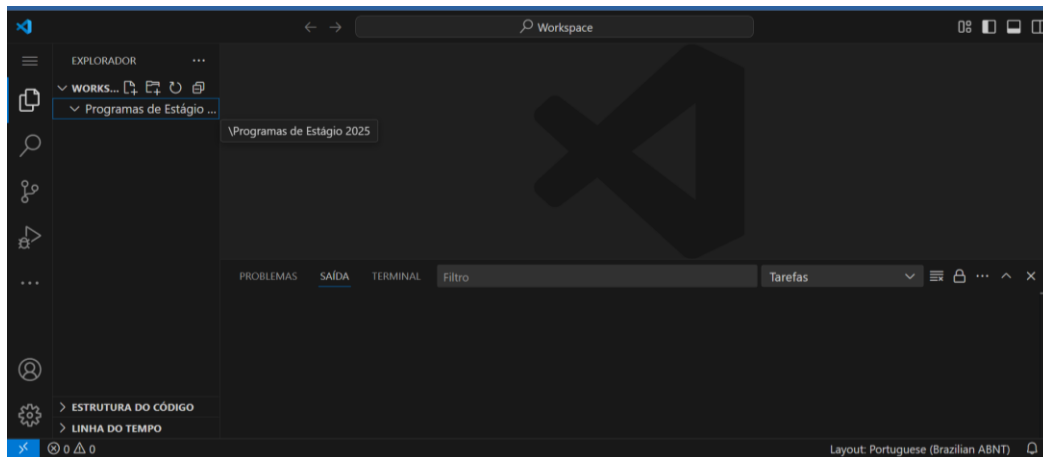
2) Abra o VS Code Online e clique em “Abrir Pasta”:



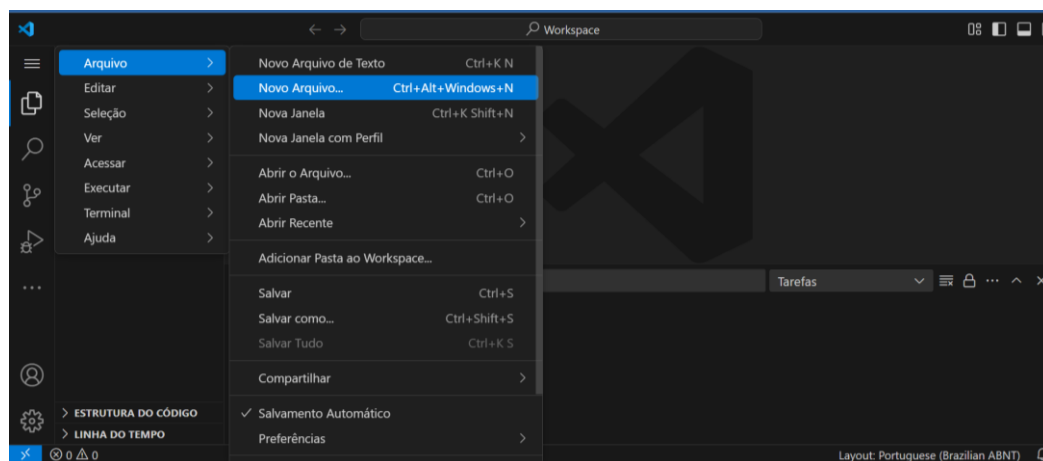
3) Selecione a pasta “Programas da Aula de Estágio” que você criou e clique em “selecionar pasta”:



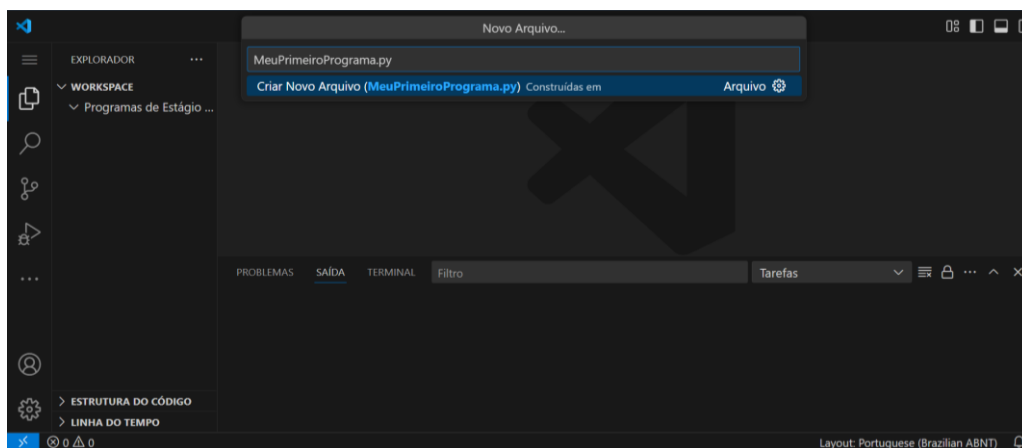
4) A pasta será aberta no VS, a pasta git estará oculta e você terá seu ambiente preparado para editar e salvar seus programas:



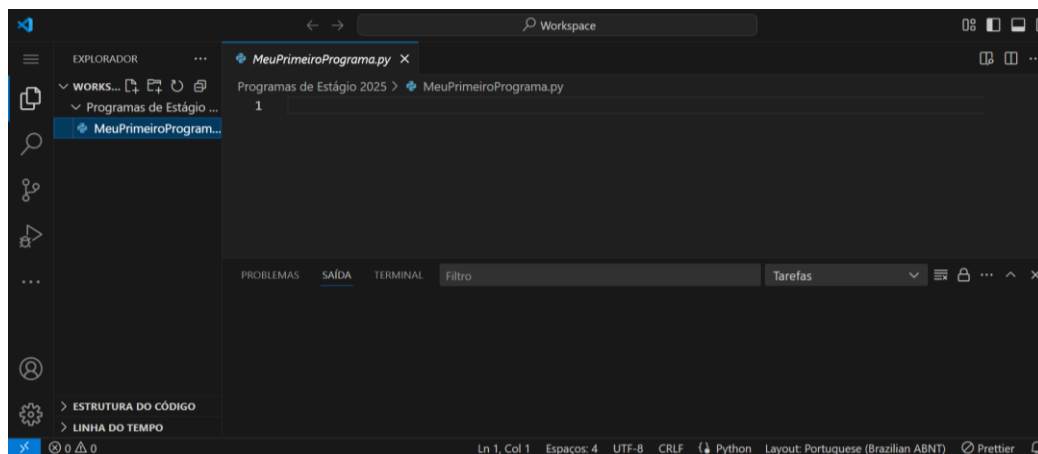
5) Clique nas três barrinhas do lado superior esquerdo da janela, se direcione para arquivo > novo arquivo e clique:



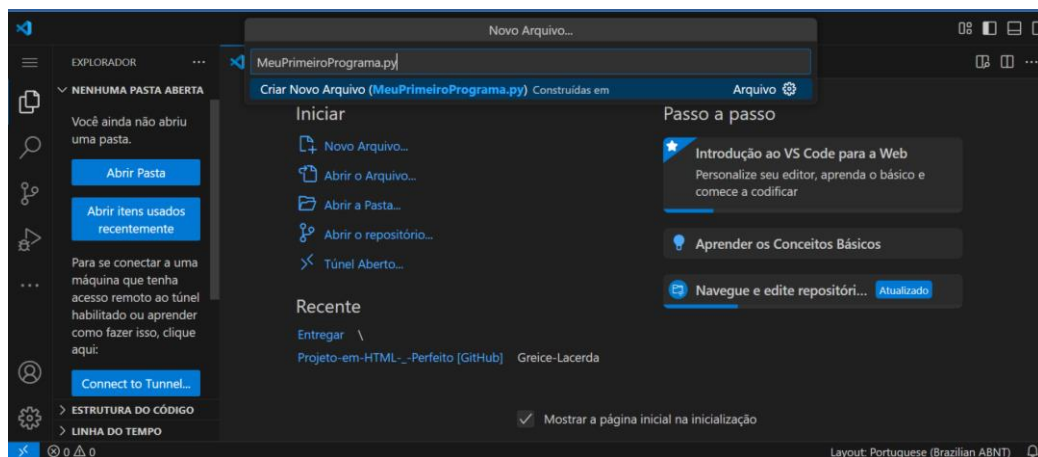
6) Crie o arquivo “MeuPrimeiroPrograma.py” como descrito na imagem abaixo, salve na pasta criada e você verá na lateral da janela, no “WORKSPACE”, sua pasta com seu arquivo:



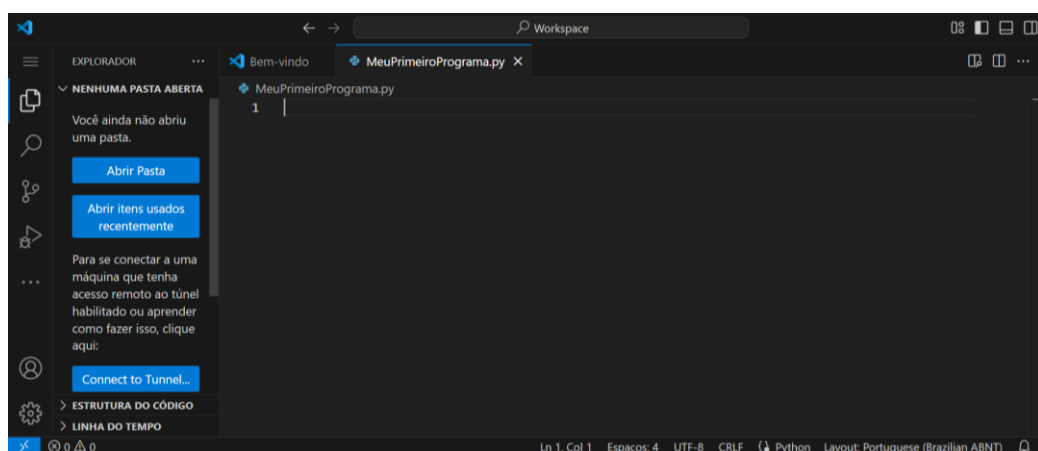
- 7) Clique no programa e você terá um ambiente preparado para inserir seu primeiro código:



- 8) Digite o nome “MeuPrimeiroPrograma.py” e clique em “Enter”:

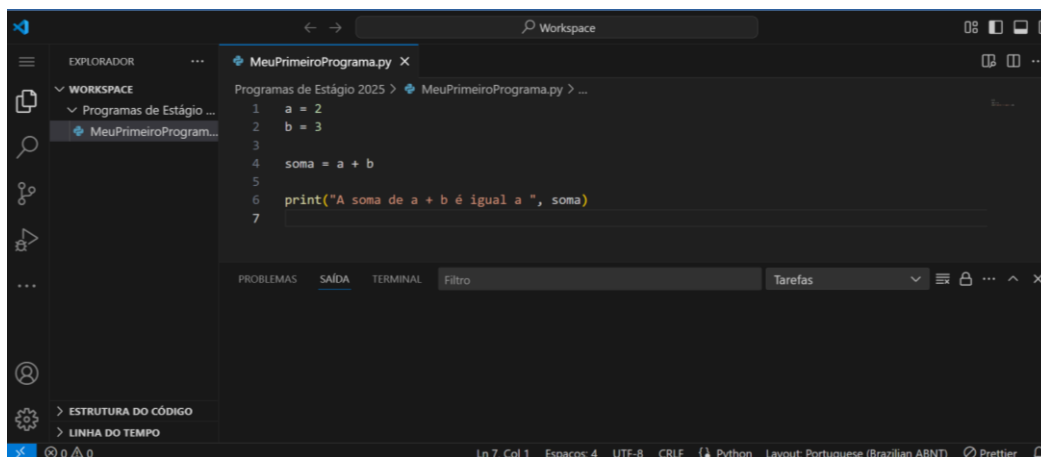


- 9) Salve seu programa na pasta que desejar e abrirá a seguinte janela, onde você digitará seu código em linguagem Python:

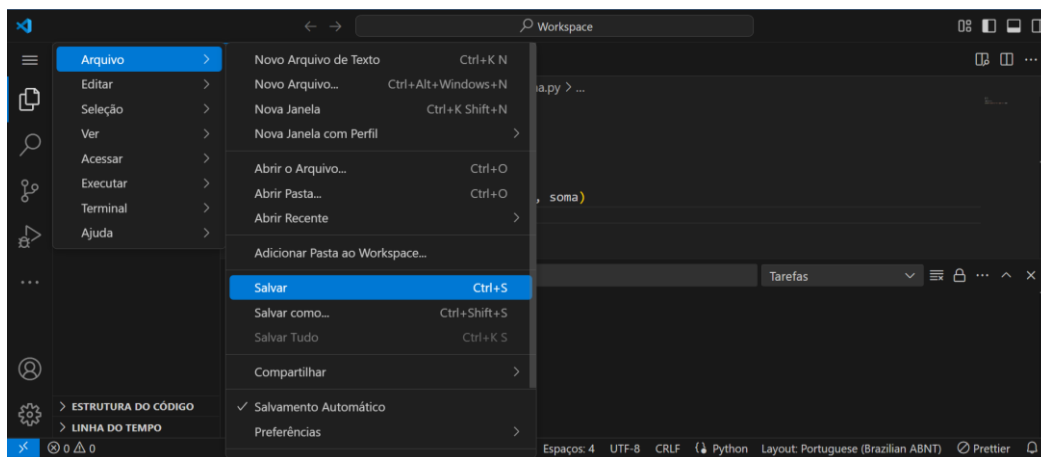


10) Digite no editor o seguinte código:

```
a = 5
b = 6
soma = a + b
print("\nA soma de a + b é igual a", soma, ".")
```

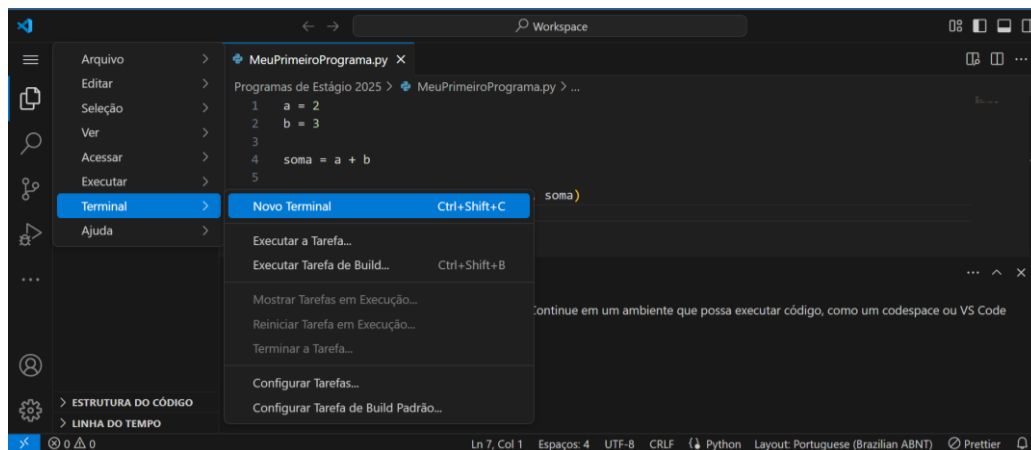


11) Clique nas três barrinhas, e vá para arquivos> salvar ou, simplesmente, clique em “CTRL + S”, para salvar seu código:



## 12) O que esse código vai gerar como resultado?

**Observação:** o VS Code online não compila o código no seu terminal, então, um pouco mais para frente nesse material, vamos conhecer uma ou duas outras ferramentas para compilar (“rodar” ou “ler”) o código: o **GBD Online** e **Codespace do GitHub**.



### 5.3. O que é o Python?

O Python é uma das linguagens de programação mais queridas e utilizadas no mundo atualmente. Em essência, esse programa é uma linguagem de programação de alto nível, interpretada, de propósito geral e com uma filosofia de design que enfatiza a legibilidade do código. Isso significa que o código Python tende a ser mais fácil de ler e escrever em comparação com muitas outras linguagens. Aqui estão alguns pontos chave sobre o Python:

- **Simples e Legível:** A sintaxe do Python é projetada para ser clara e concisa, quase como pseudocódigo. Isso facilita o aprendizado para iniciantes e torna o código mais fácil de manter e entender para equipes grandes. A indentação (espaços em branco) é crucial em Python, pois define a estrutura do código, o que força uma formatação consistente.
- **Interpretada:** Ao contrário de linguagens compiladas como C++ ou Java, o código Python é executado linha por linha por um interpretador. Isso torna o desenvolvimento mais rápido, pois você não precisa passar por um processo de compilação antes de executar o código.
- **Tipagem Dinâmica:** Você não precisa declarar explicitamente o tipo de uma variável em Python. O interpretador infere o tipo em tempo de execução. Isso pode agilizar o desenvolvimento, mas requer cuidado para evitar erros de tipo em tempo de execução.



- **Grande Biblioteca Padrão:** Python vem com uma vasta biblioteca padrão que oferece módulos para diversas tarefas, como manipulação de arquivos, redes, expressões regulares, sistemas operacionais e muito mais. Isso economiza muito tempo de desenvolvimento, pois muitas funcionalidades já estão prontas para uso.
- **Extensa Comunidade e Ecossistema:** Python possui uma comunidade de desenvolvedores enorme e ativa. Isso significa que há muitos recursos online, tutoriais, fóruns e bibliotecas de terceiros disponíveis. Se você tiver um problema, é muito provável que alguém já o tenha enfrentado e compartilhado a solução.
- **Versátil e de Propósito Geral:** Python não está limitado a um domínio específico. É usado em uma ampla variedade de aplicações, incluindo:
  - **Desenvolvimento Web:** Frameworks como Django e Flask tornam Python uma ótima escolha para construir aplicativos web robustos.
  - **Ciência de Dados e Machine Learning:** Bibliotecas como NumPy, Pandas, Scikit-learn e TensorFlow são pilares no mundo da análise de dados e inteligência artificial.
  - **Automação de Tarefas (Scripts):** Python é excelente para escrever scripts que automatizam tarefas repetitivas em sistemas operacionais.
  - **Desenvolvimento de Jogos:** Bibliotecas como Pygame permitem a criação de jogos 2D.
  - **Desenvolvimento de Desktop:** Embora não seja sua principal força, frameworks como Tkinter e PyQt permitem a criação de aplicativos de desktop.
  - **Testes de Software:** Python é usado para escrever testes automatizados.
  - **Educação:** Sua sintaxe clara o torna uma ótima linguagem para ensinar programação.
- **Orientada a Objetos:** Python suporta os princípios da programação orientada a objetos (POO), como classes, objetos, herança, polimorfismo e encapsulamento, permitindo a criação de código modular e reutilizável.
- **Multiplataforma:** Python roda em diversos sistemas operacionais, como Windows, macOS e Linux, sem grandes modificações no código.

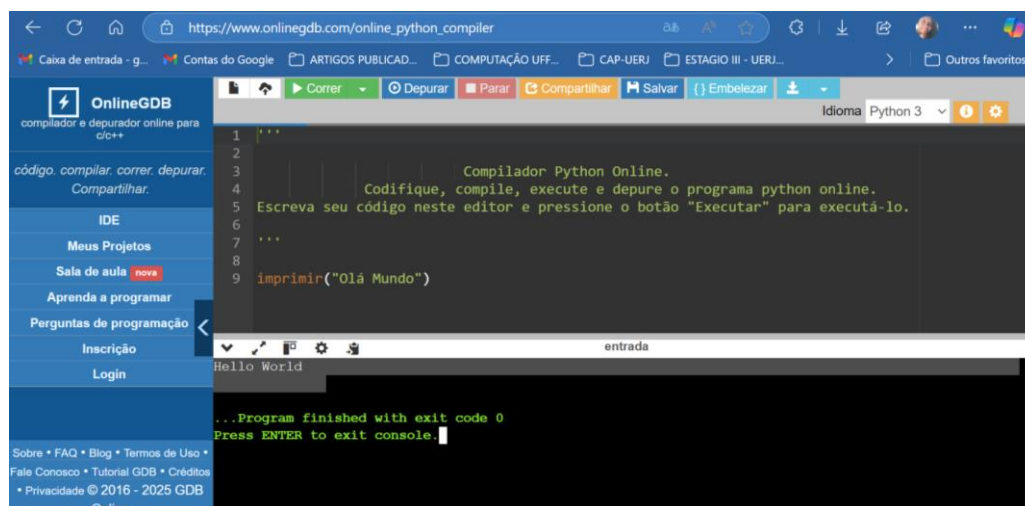
Em resumo, Python é uma linguagem poderosa, flexível e fácil de aprender que se tornou essencial em muitos campos da tecnologia. Sua legibilidade, vasta biblioteca e forte comunidade a tornam uma excelente escolha tanto para iniciantes quanto para desenvolvedores experientes.

Contudo, nosso foco nesse curso não será o estudo do Python. Mas, o utilizaremos em alguns momentos como ferramenta de apoio para os conhecimentos a serem desenvolvidos. Por ora, vamos retornar ao GitHub.

## 5.4. O que é o GitHub?

O **GitHub** é uma plataforma de hospedagem de código que permite aos desenvolvedores armazenar, gerenciar e colaborar em projetos de software. Ele utiliza o sistema de controle de versão Git, que facilita o rastreamento de alterações no código e a colaboração entre equipes. E também possui o “**Codespace**” que possibilita a compilação de seu código de forma online.

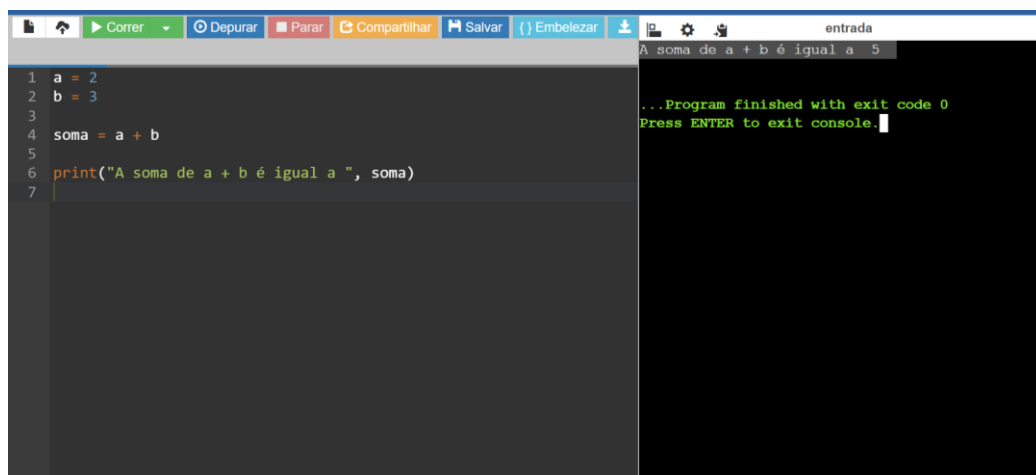
Ou ainda, podemos usar um compilador online “GBD COMPILER PYTHON ONLINE”, no link: [Online Python Compiler - editor online](https://www.onlinegdb.com/online_python_compiler), para visualizar o resultado do programa. Veja a janela do compilador:



Nesse compilador, você também pode salvar seu projeto e compilar muitas outras linguagens de programação. Ressalto que escolhi os editores e compiladores online para facilitar a prática, sem precisar baixar e instalar programas.

Mas, você pode baixar e instalá-los no computador, se quiser, e ter acesso todas as suas ferramentas. Por exemplo, como VS Code no desktop, não é necessário usar o GBD Online, pois o código pode ser compilado no terminal do VS Code.

Visualizando o resultado do código: copie o código digitado no VS Code Online e cole no GBD Online. Clique em “RUN” ou “CORRER” e verifique na janela o resultado:



The screenshot shows the VS Code Online interface. The editor on the left contains the following Python code:

```
1 a = 2
2 b = 3
3
4 soma = a + b
5
6 print("A soma de a + b é igual a ", soma)
7
```

The terminal on the right shows the output of the program:

```
A soma de a + b é igual a 5
...Program finished with exit code 0
Press ENTER to exit console.
```

Mas, retornando ao nosso programa de versionamento o GITHUB, podemos dizer que algumas das suas principais funcionalidades do GitHub são:

- **Repositórios:** armazene e organize seus projetos de código.
- **Controle de Versão:** use-o para rastrear alterações e gerenciar versões do código.
- **Colaboração:** trabalhe com outros desenvolvedores através de pull requests, issues e revisões de código.
- **Integrações:** conecte-se a outras ferramentas de desenvolvimento para automatizar fluxos de trabalho.
- **GitHub Pages:** hospede sites estáticos diretamente a partir de um repositório GitHub.

#### 5.4.1. Criação de Conta e Configuração Inicial

Vejamos como criar a sua conta no GitHub:

1. **Acesse o GitHub:** Vá para [GitHub](https://github.com).
2. **Inscreva-se:** clique em "Sign up" e preencha os campos necessários (nome de usuário, e-mail e senha).
3. **Verificação de E-mail:** verifique seu e-mail para confirmar a criação da conta

4. **Configuração Inicial:** complete as informações do perfil e configure a autenticação de dois fatores (2FA) para maior segurança

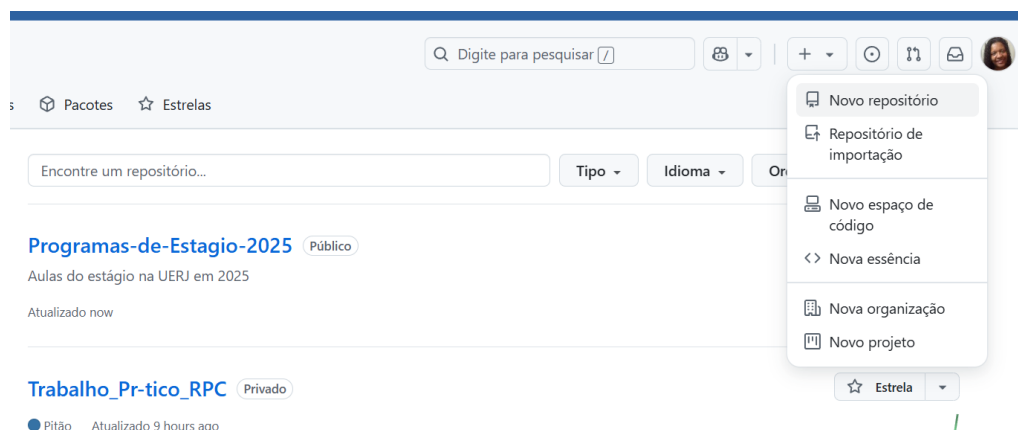
#### 5.4.2. Links para pesquisa que podem ajudar com as configurações do GitHub:

- [Criar uma conta no GitHub](#)
- [Introdução à sua conta do GitHub](#)
- [Lista de comandos úteis do GIT](#)
- [Os 20 principais comandos do Git com exemplos](#)
- [Introduction to Git in VS Code](#)
- [Clonar um repositório - GitHub Docs](#)

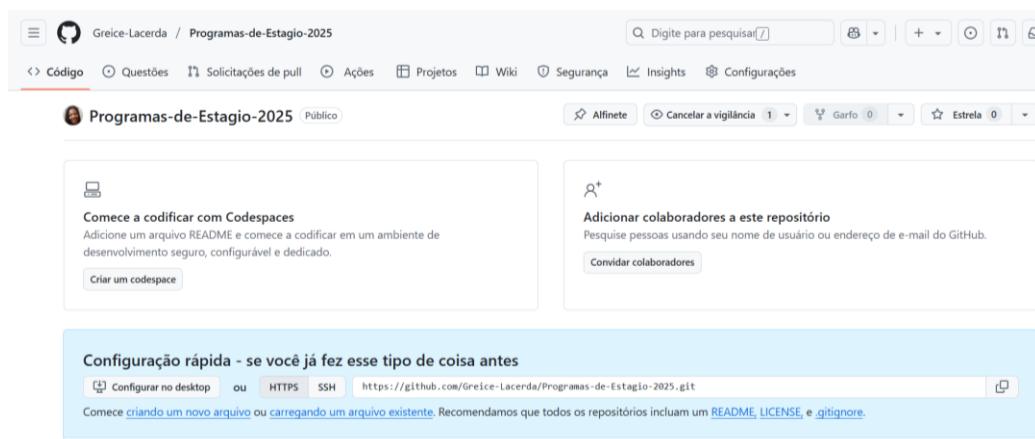
#### 5.4.3. Iniciando um repositório Git para as aulas de estágio

Agora que você conhece o VS Code Online e o GitHub, vamos iniciar um repositório? Siga os passos a abaixo:

1) Na sua conta Git, crie um novo repositório clicando no sinal de “+” na janela, no canto superior direito e escolha um nome. Veja a imagem:



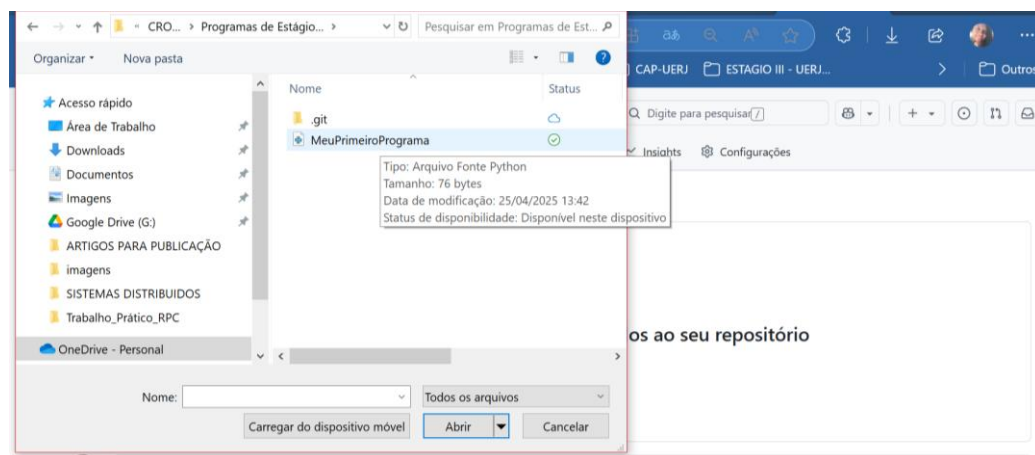
2) No seu repositório, na barra azul no fim da página, clique em “carregando um arquivo existente”:



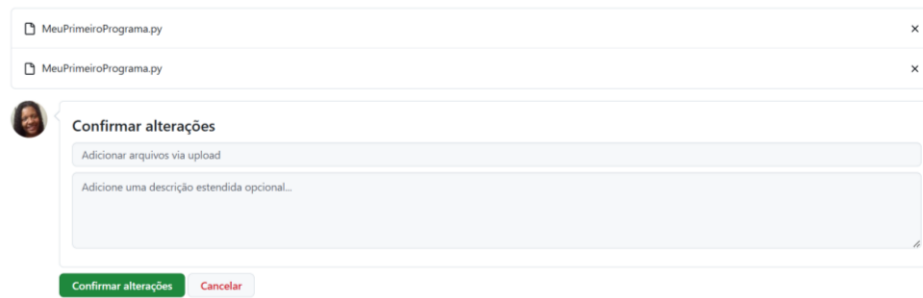
### 3) Escolha o arquivo a ser adicionado:



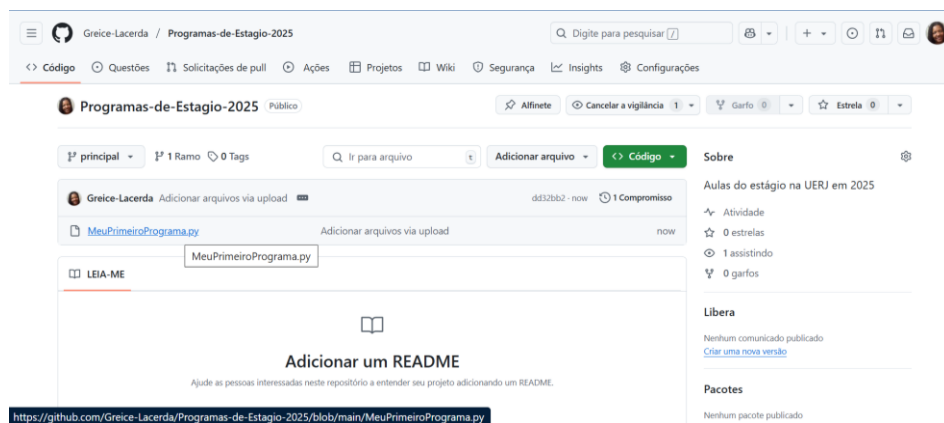
### 4) Selecione o arquivo da aula: “MeuPrimeiroPrograma” e em Abrir:



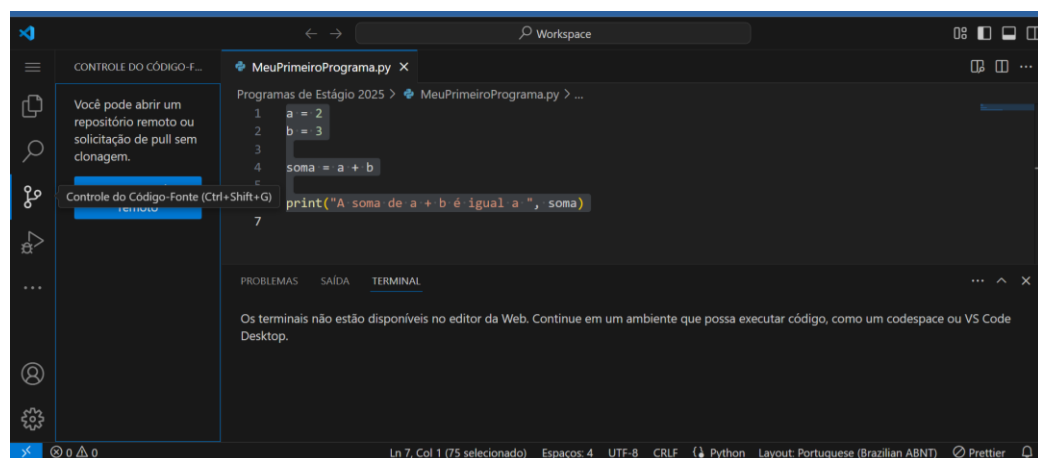
5) Seu arquivo será enviado para o upload no Git e você pode adicionar uma descrição sobre ele, se quiser. Clique em “Confirmar alterações”



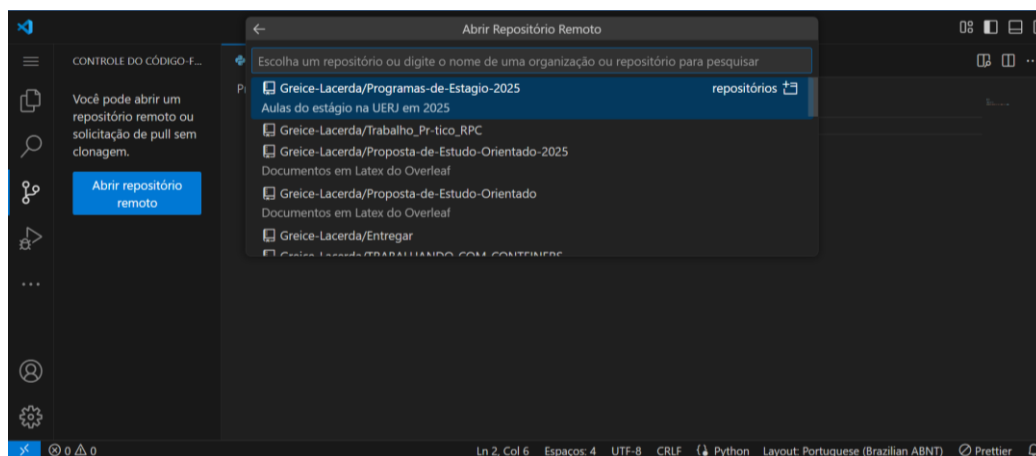
6) Pronto! Seu primeiro arquivo foi salvo no GitHub:



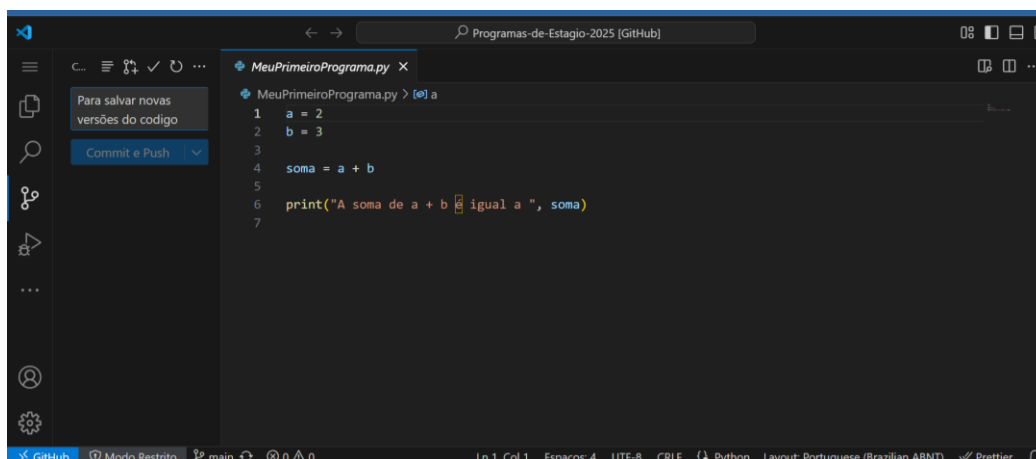
7) No VS Code Online e navegue até a aba "Source Control" ou “Controle de código-fonte”. Veja na imagem:



8) Clique em "Abrir um repositório remoto". Você vai precisar “logar” com a sua conta Git. Selecione o repositório criado e seu arquivo será “puxado” do Git para o VS e estará pronto para salvar novas versões. Veja:



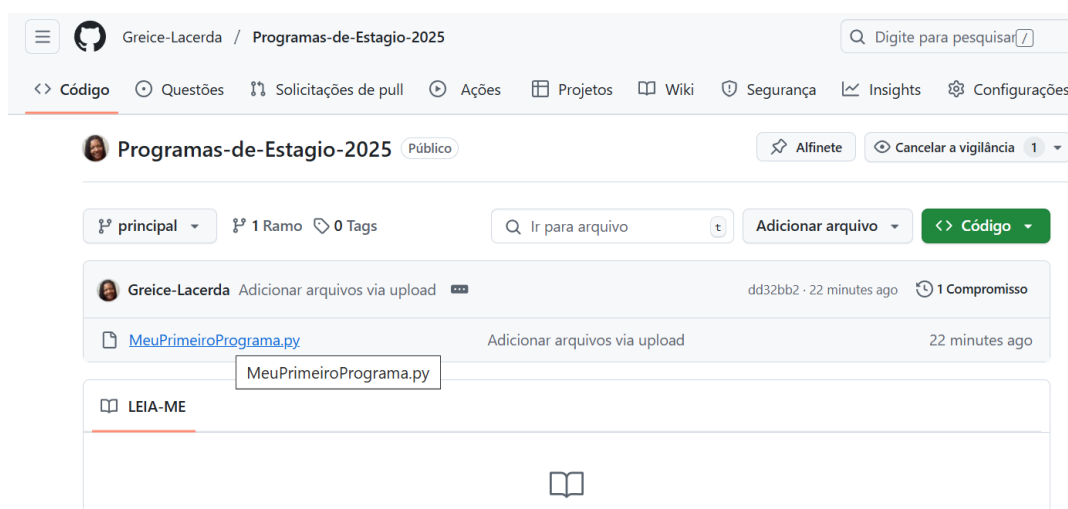
9) Clicando no arquivo, você verá no “WORKSPACE”, a janela de versionamento. E na barra inferior esquerdo, abaixo de configurações, as setinhas  $\gg$ , agora apontam para GitHub. Verifique:



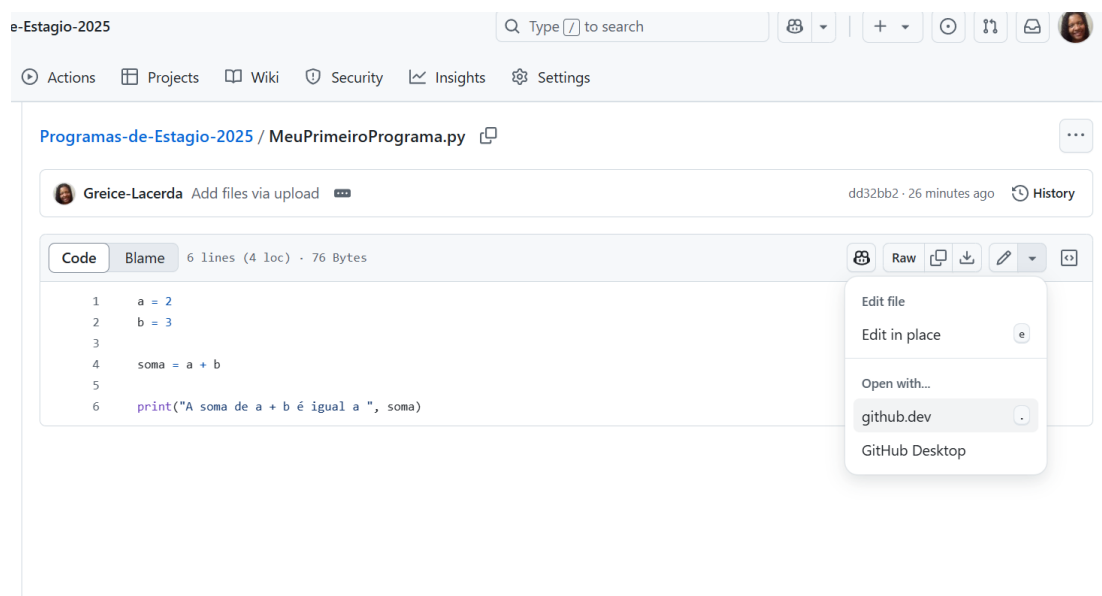
#### 5.4.4. Usando o Codespace do GitHub

Vamos usar o repositório para visualizar os resultados gerados pelo código? Siga os passos abaixo:

1) Agora, retorne para a janela do GitHub no navegador e clique no seu arquivo.

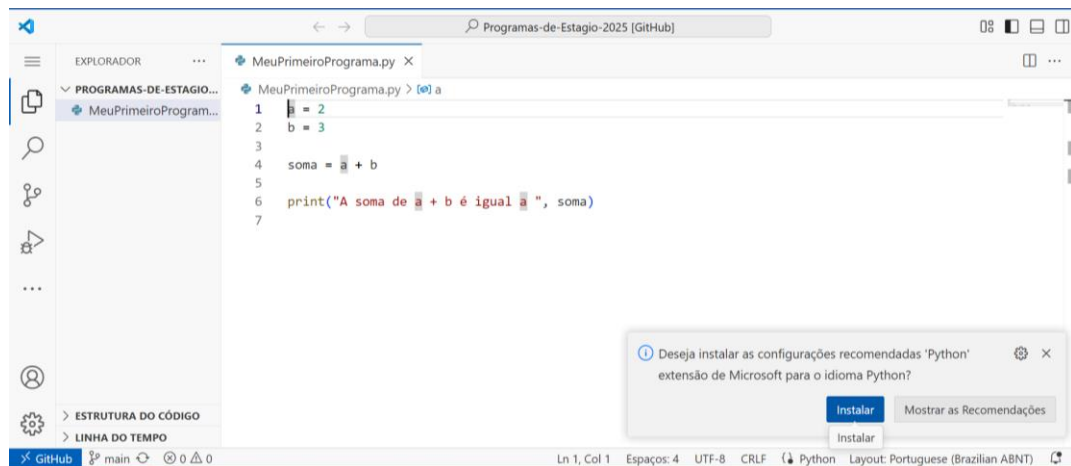


2) No ícone do lápis no lado direito da tela, clique na setinha e selecione “github.dev”:

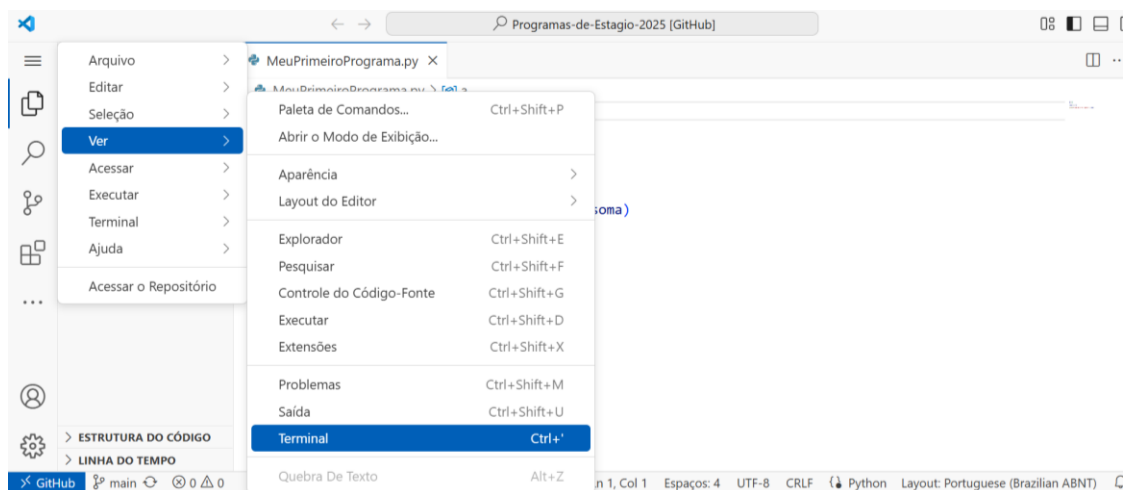


3) Você terá acesso ao codespace com o seu código, será necessário aceitar instalar a extensão python. Observe que o codespace é um VS Code dentro do GitHub.

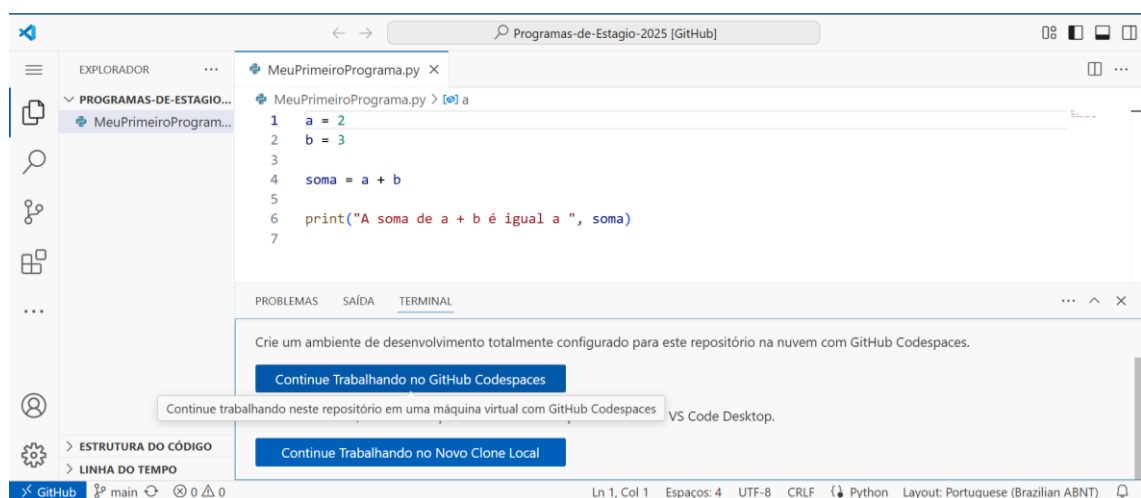




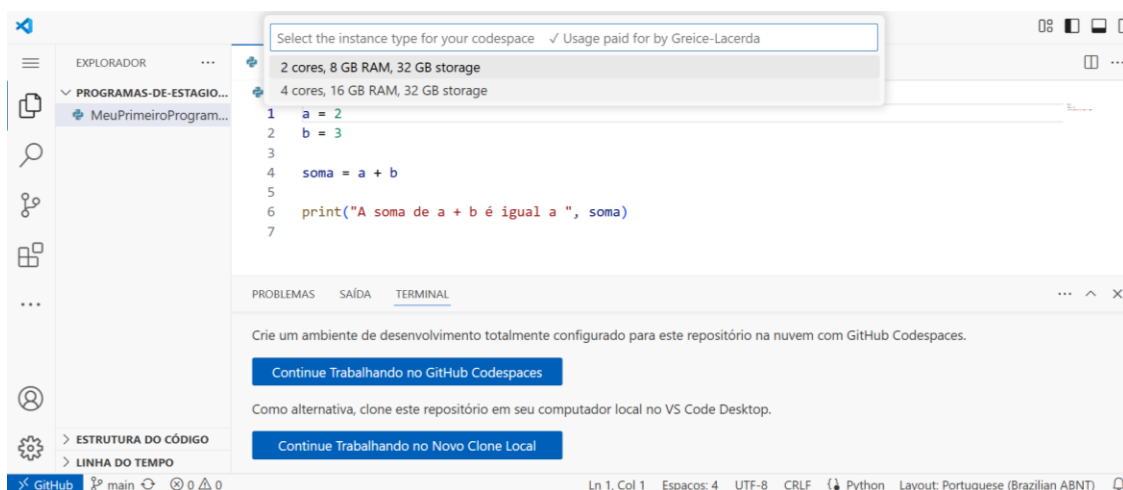
4) Siga para as três barrinhas > ver > terminal:



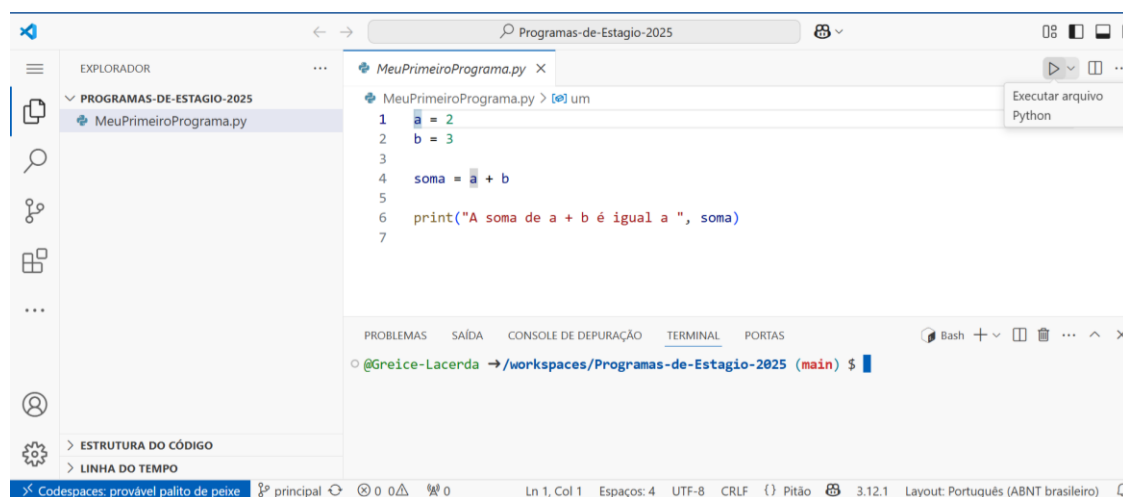
5) Escolha “Continuar Trabalhando no GitHub Codespace”:



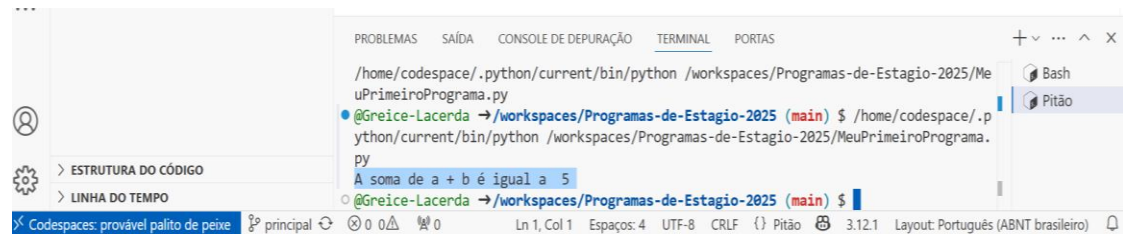
6) Selecione o tipo do seu codespece: 2 ou 4 cores:



7) Clique no canto superior direito na setinha “RUN”, descrita: “Executar arquivo Python”:



8) Observe a compilação do programa no terminal do Codespace:

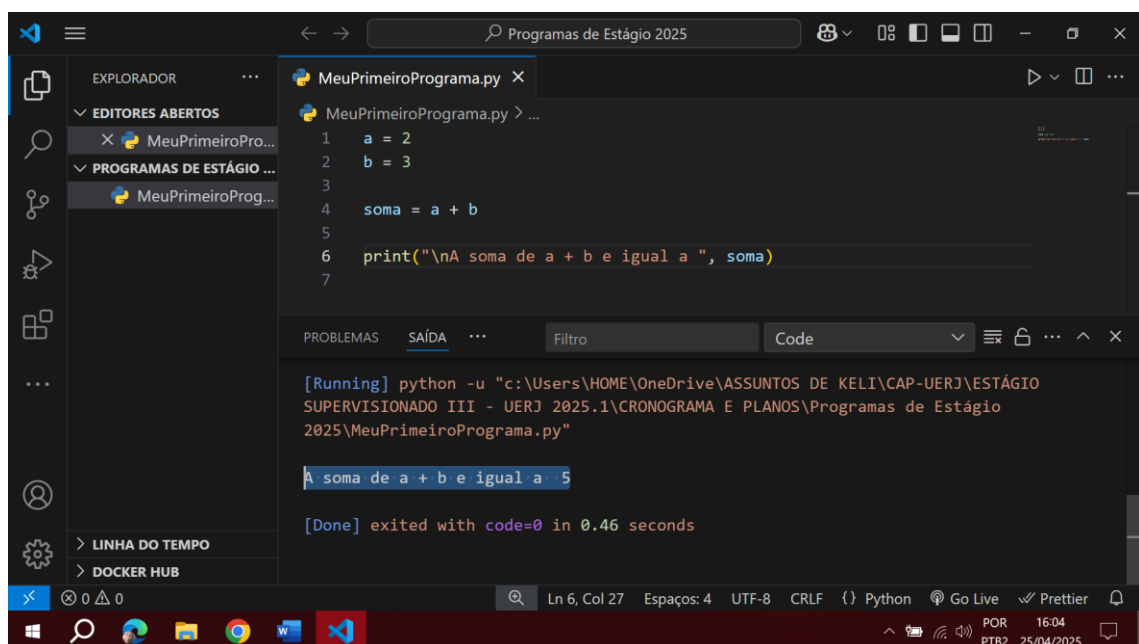


9) Agora, você está iniciando no mundo do desenvolvimento! Mas, **qual a diferença do VS Code Online para o Codespace do GitHub?**

- Pense no **VS Code Online** como um **editor de código web leve e conveniente** para visualização e edições rápidas, sem a necessidade de instalação local e sem capacidade de execução nativa.
- Considere o **GitHub Codespace** como um **ambiente de desenvolvimento completo e poderoso na nuvem**, que inclui um editor VS Code, um sistema operacional e recursos para executar, construir e depurar seus projetos, com forte integração com o GitHub e um custo associado.

Então, se você precisa apenas visualizar ou fazer pequenas alterações em arquivos online, o VS Code Online pode ser suficiente. Se você precisa de um ambiente de desenvolvimento funcional na nuvem para trabalhar em projetos, executar testes e construir aplicações, o GitHub Codespace é a solução mais adequada.

Agora, se você quer criar seus códigos, compilar, verificar seu funcionamento e salvar suas versões enviando-as para o GitHub. Tudo em um só ambiente! Eu sugiro usar o VS Code Desktop. A escolha é sua! Veja uma imagem do programa:



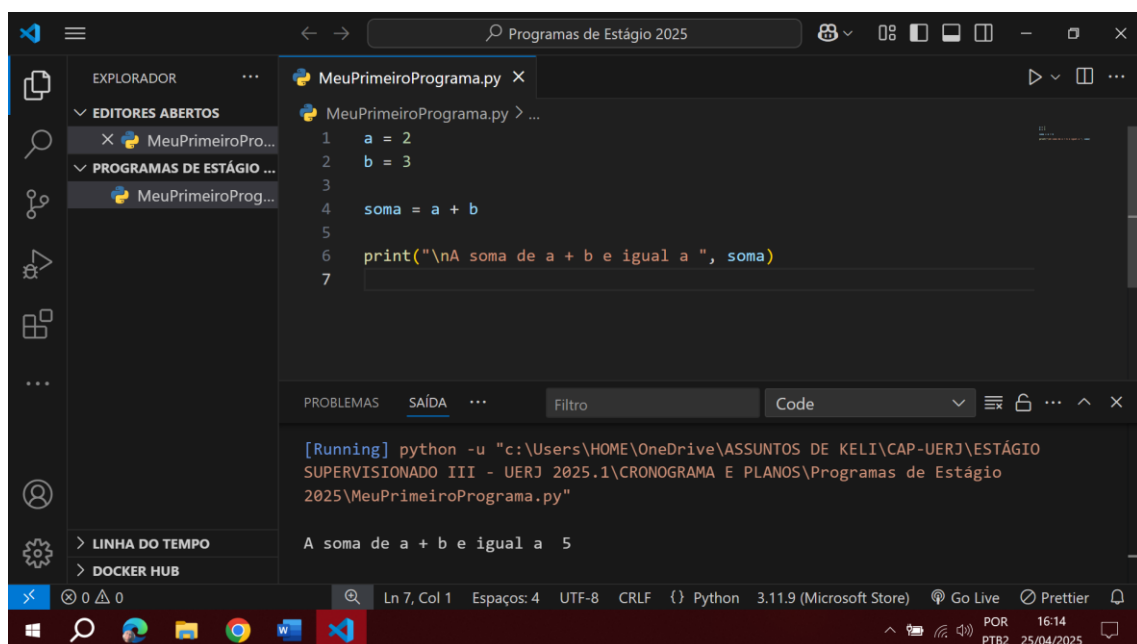
## 6. Momento 2: Conflitos Cognitivos e a Natureza dos Dados

**Foco:** Utilizar problemas matemáticos simples para provocar conflitos cognitivos, forçando uma reflexão sobre a diferença entre a representação matemática e a computacional.

### 6.1. Vamos prática com um pouco mais de matemática?

Escolha em qual ambiente de desenvolvimento quer trabalhar: **VS Code Online**, **Codespace no Git**, **VS Code Desktop** ou **GDB Online**. E retorne para o navegador com o ambiente escolhido. Se desejar, pode fechar os outros navegadores abertos. Os exemplos que se seguirão foram elaborados no VS Code Desktop.

#### a) Retorne ao código digitado:



```
MeuPrimeiroPrograma.py
1  a = 2
2  b = 3
3
4  soma = a + b
5
6  print("\nA soma de a + b e igual a ", soma)
7
```

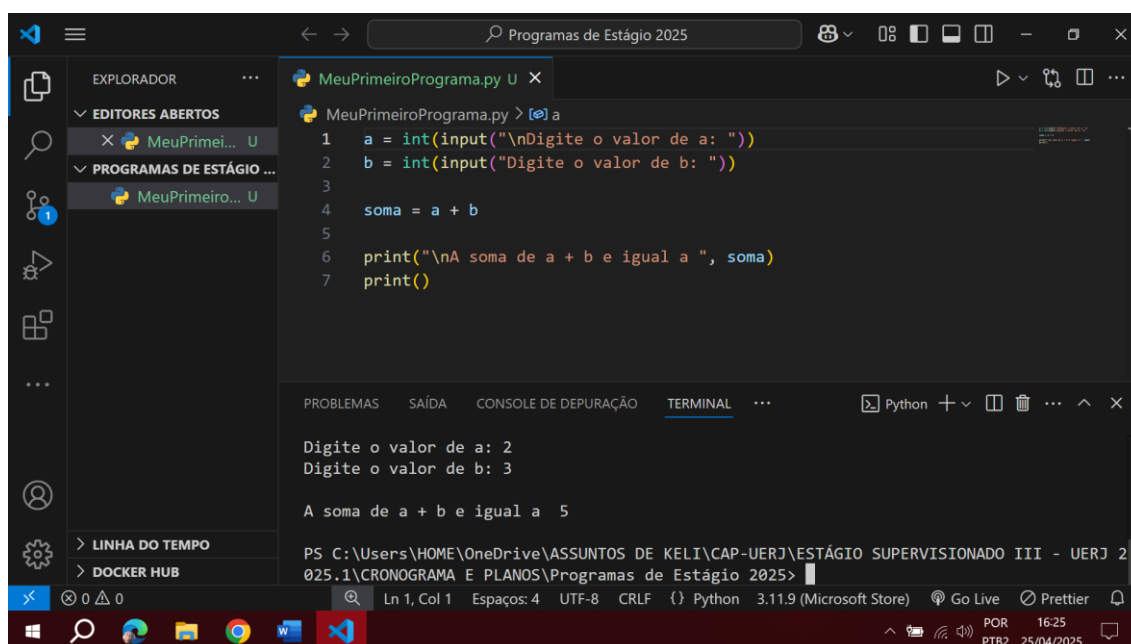
PROBLEMAS SAÍDA ... Filtro Code

[Running] python -u "c:\Users\HOME\OneDrive\ASSUNTOS DE KELI\CAP-UERJ\ESTÁGIO SUPERVISIONADO III - UERJ 2025.1\CRONOGRAMA E PLANOS\Programas de Estágio 2025\MeuPrimeiroPrograma.py"

A soma de a + b e igual a 5

#### b) Altere as seguintes linhas no seu código:

```
a = int(input("Digite o valor de a: "))
b = int(input("Digite o valor de b: "))
```



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a file named 'MeuPrimeiroPrograma.py'. The Editor pane displays the following Python code:

```
1 a = int(input("\nDigite o valor de a: "))
2 b = int(input("Digite o valor de b: "))
3
4 soma = a + b
5
6 print("\nA soma de a + b e igual a ", soma)
7 print()
```

The TERMINAL pane at the bottom shows the output of the program:

```
Digite o valor de a: 2
Digite o valor de b: 3

A soma de a + b e igual a 5
```

The terminal prompt is 'PS C:\Users\HOME\OneDrive\ASSUNTOS DE KELI\CAP-UERJ\ESTÁGIO SUPERVISIONADO III - UERJ 2025.1\CRONOGRAMA E PLANOS\Programas de Estágio 2025>'.

**O que foi apresentado no terminal? Explique por quê?**

- Salve essa versão do programa no GitHub, com o “commit: Valores inteiros”.

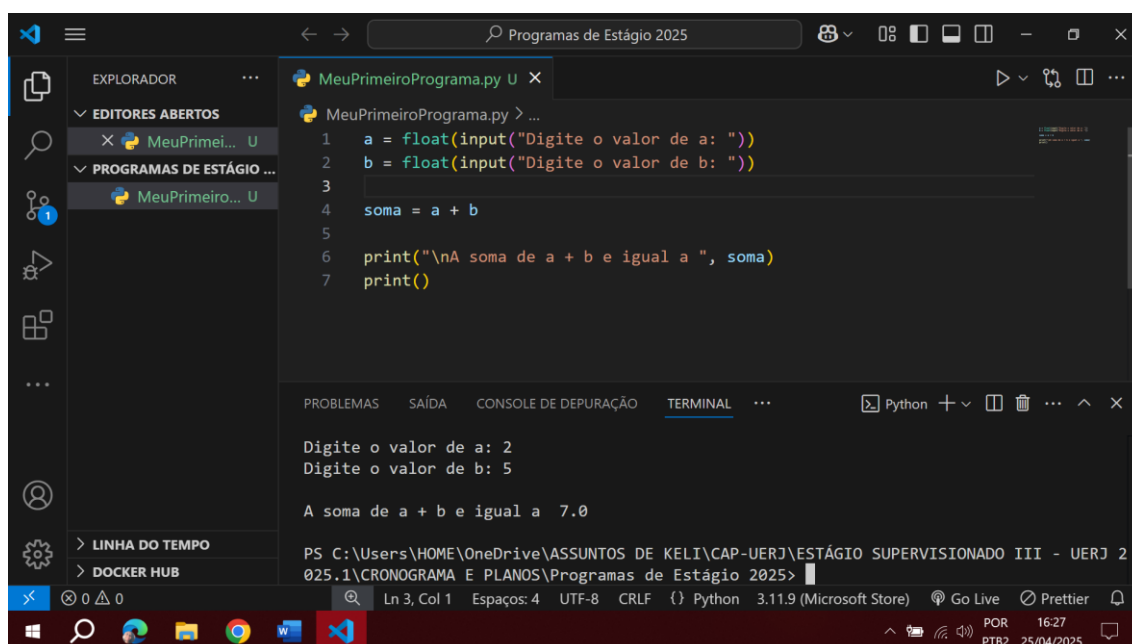
**d) Agora, altere as linhas no seu código para:**

```
a = float(input("Digite o valor de a: "))
```

```
b = float(input("Digite o valor de b: "))
```

**Qual a diferença do primeiro resultado para esse resultado? Explique essa diferença.**

Veja o resultado da compilação:



- Salve essa versão do programa no GitHub, com o “commit: Valores Reais”.

**e) Altere novamente as linhas no seu código para:**

```
a = input("Digite o valor de a: ")
```

```
b = input("Digite o valor de b: ")
```

Responda:

1. O que foi impresso na saída do programa?
  2. O resultado apresentado é um erro de compilação?
  3. Como você explica esse resultado.
  4. Observe a imagem e compare-a com as imagens dos outros resultados e com o código que gerou cada uma delas, o que mudou?
- Salve essa versão do programa no GitHub, com o “commit: Strings”.
  - O que é uma string?

**6.2. Agora, experimente outras operações básicas consultando o link: [Python: Guia de Operadores - DevMedia](#).**

## Curiosidades!

A DevMedia é uma plataforma que visa impulsionar a carreira de desenvolvedores através de diversos recursos educacionais. Eles oferecem artigos informativos sobre variados temas da programação, como o artigo que explica os operadores na linguagem Python.

## Agora é com você!

### 7. Momento 3: Aplicação, Autonomia e Pensamento Escalável

**Foco:** *Propor um problema prático mais complexo para que os licenciandos apliquem os conceitos aprendidos e comecem a pensar em termos de eficiência e generalização.*

Experimente os exemplos da plataforma DevMedia e na próxima aula apresente dois programas distintos em Python para:

- a) solicitar a entrada de cinco números reais, calcular e imprimir a média aritmética desses números.
- b) solicitar a entrada do valor da medida do raio  $r$  de um círculo qualquer e, usando a biblioteca “*import math*”, calcular e imprimir o valor de seu perímetro e de sua área.

## Experimente!

- Média aritmética, sem o comando “for”, acesse: <https://onlinegdb.com/rW5wTgbNT>
- Média com comando “for”, acesse: <https://onlinegdb.com/Evl7vwJzd>
- Perímetro e área de um círculo qualquer, acesse: <https://onlinegdb.com/8z6BBYnyt>