

## Usando VSCode com Python:

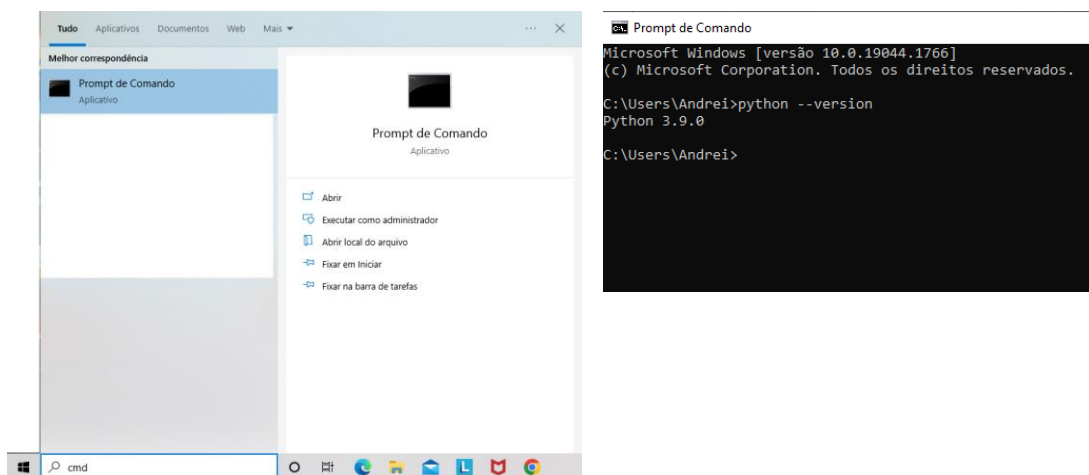
O Visual Studio Code é um editor de código extensível desenvolvido pela Microsoft que pode ser usado como ferramenta auxiliar na programação em diferentes linguagens tais como Python, C/C++, Java, JavaScript, PHP e outras. Ele também serve para editar arquivos de texto em diferentes formatos tais como TXT, CSV, JSON, XML, HTML, CSS, e outros.

Algumas de suas vantagens são o fato de ser leve, poder ser instalado/utilizado nos mais diversos sistemas operacionais e dispositivos, rodar e depurar (testar/avaliar) códigos de programas em diversas linguagens, auto completar/corrigir/sugerir códigos, integrar com outras ferramentas como Git e GitHub, é de código livre e gratuito.

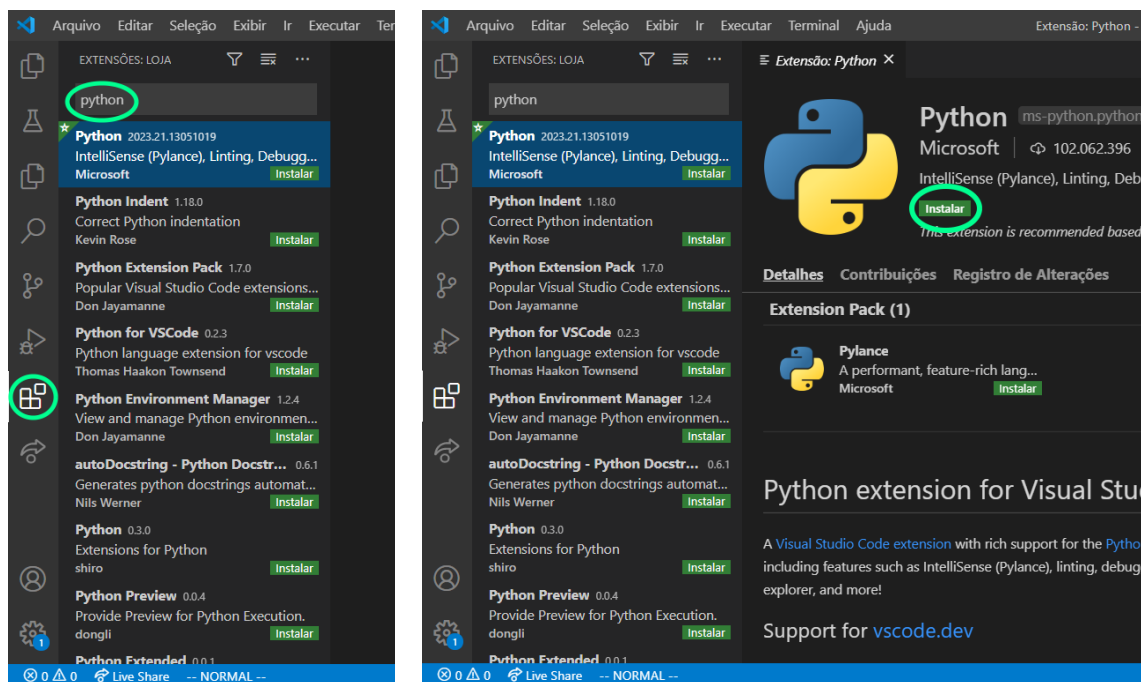
Para Python, é uma das ferramentas mais utilizadas, perdendo somente para o PyCharm em popularidade. O PyCharm, embora mais completo em questão de recursos para auxiliar os programadores em Python, sendo exclusivo da linguagem Python, é mais pesado, complexo e proprietário (apesar de ter uma versão comunitária que é gratuita).

## Atividade VSCode:

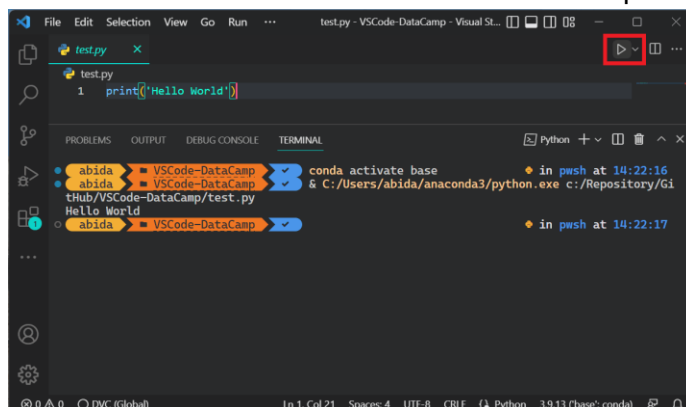
- 1) Baixe e instale o interpretador da linguagem Python.
  - a) <https://www.python.org/downloads/>
  - b) Siga as configurações padrões de instalação.
  - c) Para testar a instalação, abra o terminal/CMD (prompt de comando), digite `python --version` e pressione enter. Deverá aparecer a versão do python instalada.



- 2) Baixe e instale o Visual Studio Code (VSCode) na sua máquina.
  - a) <https://code.visualstudio.com/>
  - b) Siga as configurações padrões de instalação.
- 3) Abra o VSCode, acesse o menu *Extensões* do lado esquerdo, faça uma busca por *python* e instale a extensão *Python* da *Microsoft*.



- 4) Abra uma pasta em *Arquivo* -> *Abrir Pasta...*
- 5) Crie um novo arquivo em *Arquivo* -> *Novo Arquivo*
  - a) Salve o arquivo, nomeie com qualquer nome (sem espaços) e finalize com a extensão `.py`.
- 6) O arquivo estará em branco, basta digitar o código em Python agora.
  - a) Comece com um *Hello World* básico e clique no botão para executar.



- b) Note que uma aba chamada de terminal abrirá abaixo com a saída do programa.
  - c) Os inputs (entradas do usuário) também aparecerão no terminal do VSCode e o usuário poderá interagir/informar os valores através dessa interface/aba do terminal.
- 7) Selecione alguns dos programas que você já escreveu na ferramenta *Google Colab* utilizada nas aulas e execute os programas selecionados no VSCode.
- 8) Faça uma captura de tela da execução de um dos programas selecionados no passo anterior e submeta a imagem através da atividade correspondente no SIGAA.

Se quiser ver um conteúdo adicional sobre o uso do VSCode para Python, você pode consultar o seguinte sítio, que também serviu de fonte para as imagens utilizadas: <https://www.datacamp.com/tutorial/setting-up-vscode-python>

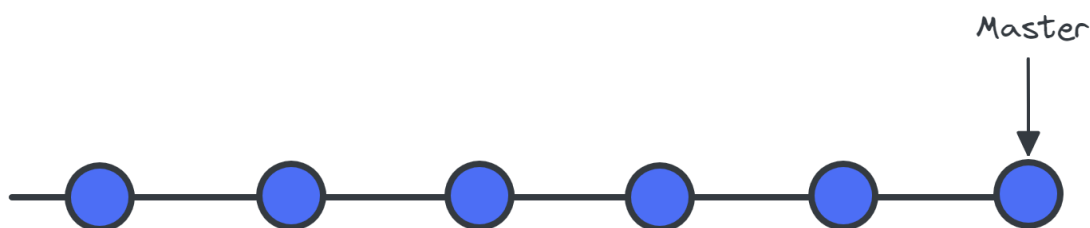
Agora você já tem tudo o que precisa para criar e executar os programas em Python na própria máquina, mas muitas vezes você precisará trabalhar em grupos, controlar versões e publicar/disponibilizar o seu código/programa para outras pessoas.

### Usando Git, GitHub com o VSCode:

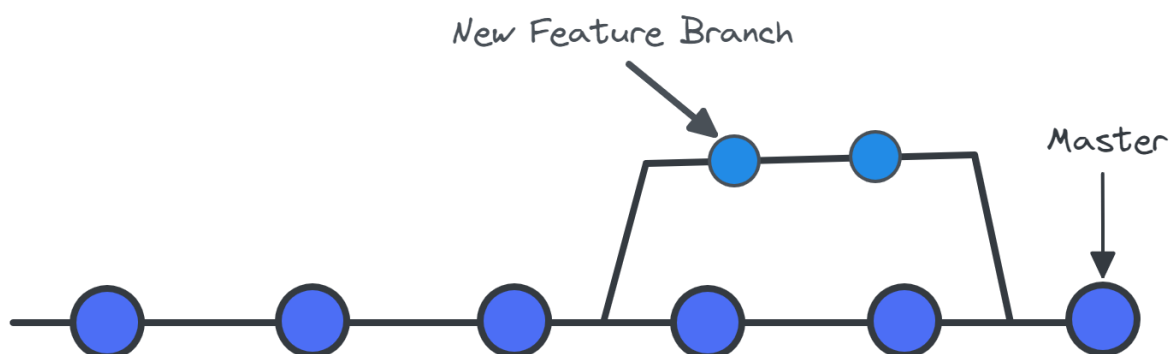
Git é um sistema de controle de versão distribuído de arquivos de código aberto usado principalmente no desenvolvimento de sistemas. O sistema Git hoje é o sistema de controle de versão mais utilizado no mundo, ultrapassando em uso o tradicional sistema SVN (Apache Subversion).

O Git armazena e gerencia um projeto (estruturado em pastas e arquivos) em um **repositório local** na máquina do usuário, guardando e controlando as modificações realizadas através de pontos de salvamento chamados de **commits**. Ele permite ao usuário comparar diferentes commits dentro do histórico do repositório e regredir para pontos de salvamento anteriores.

Esse histórico é armazenado em uma estrutura como a mostrada a seguir, onde os nós (círculos azuis) são os pontos de salvamento. O ramo principal dessa estrutura é chamado de **Master** (ou **Main**).



O Git também permite ao usuário criar outros ramos (**branches**) na estrutura para controlar modificações no projeto de forma paralela sem afetar o ramo principal e apenas unir os ramos (**merge**) quando as modificações estiverem prontas, estáveis ou consolidadas.

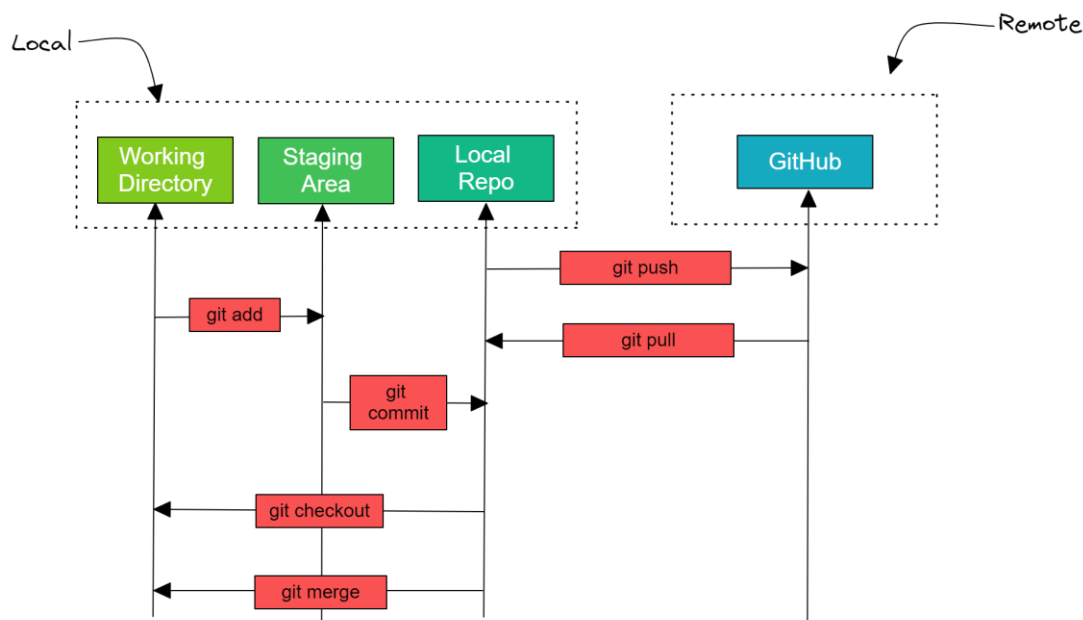


A divisão da estrutura em diversos ramos também é importante quando se deseja trabalhar em equipe na construção de um sistema/projeto grande com muitas funcionalidades. Para poder trabalhar em equipe, com múltiplos usuários atuando em um mesmo projeto, também

utiliza-se um repositório remoto que se comunica com o repositório local da máquina de cada usuário.

O repositório remoto pode ser instalado/configurado em uma máquina qualquer que tenha comunicação em rede com todas as máquinas locais usadas pelos usuários. Para essa finalidade, existe a plataforma GitHub, um sistema gratuito que disponibiliza a criação de um repositório remoto que pode ser acessado por qualquer um com acesso à internet e gerenciado pelo dono da conta vinculada.

O fluxo geral do Git entre um repositório local e um repositório remoto pode ser visualizado na imagem abaixo:

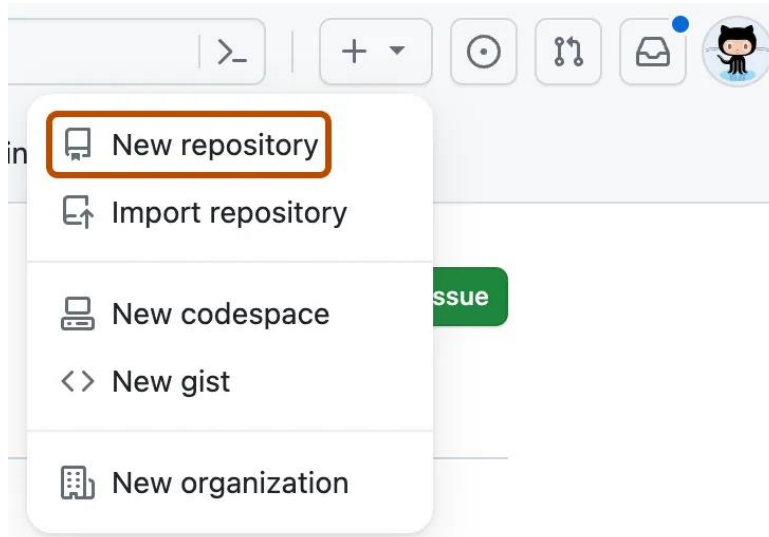


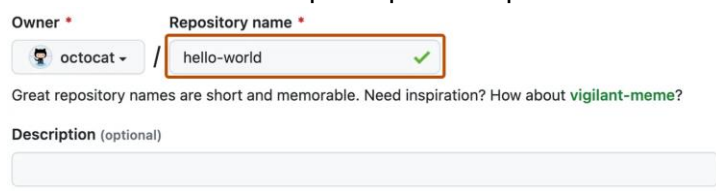
A área de trabalho (**working directory**) corresponde a cópia local dos arquivos do projeto em que o usuário está trabalhando em sua própria máquina. Antes de enviar as suas alterações no projeto para o repositório local, efetivamente guardando o histórico dessas alterações, o usuário deve adicionar (**git add**) as alterações na área de preparação (**staging area**) para então salvar (**git commit**) o histórico dessas alterações no repositório local. Para publicar o seu repositório local em um repositório remoto (como o GitHub) é necessário primeiro sincronizar ambos repositórios (**git pull**) e então realizar a submissão das modificações locais (**git push**). Por fim, quando o usuário deseja retornar a um ponto específico de salvamento dentro do histórico armazenado no seu repositório local ele pode fazer um **git checkout**. Já o **git merge** é utilizado quando se deseja manter as alterações realizadas na área de trabalho ao mesmo tempo que se pretende atualizar a versão do projeto com modificações oriundas do repositório local (muito provavelmente realizadas por outro usuário e que foram sincronizadas via repositório remoto) ou quando se deseja aplicar as alterações de um **branch** ao **master** ou vice-versa (embora no sentido contrário seja comum usar o comando específico chamando de **git rebase**).

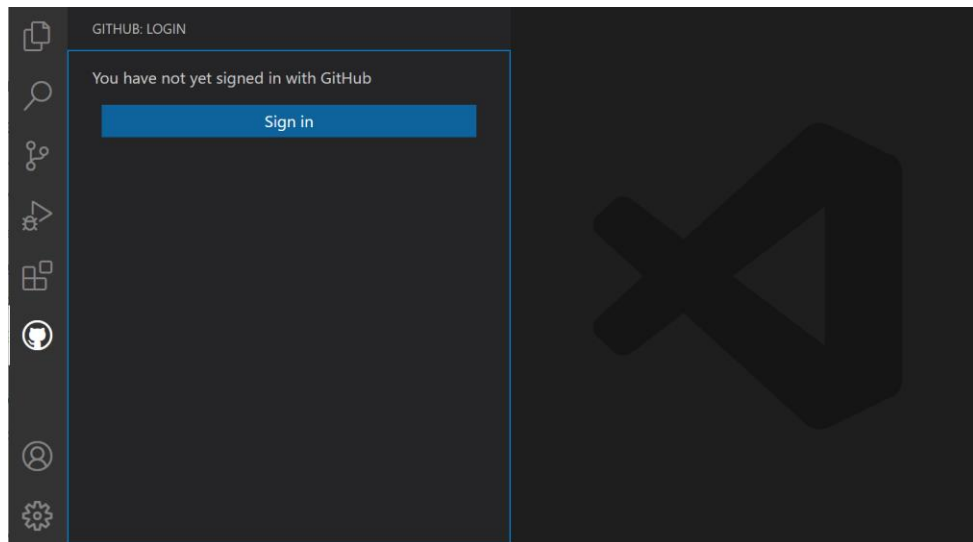
Mais informações sobre Git e para consultar fonte das imagens utilizadas, acesse o sítio: <https://www.datacamp.com/tutorial/github-and-git-tutorial-for-beginners>

## Atividade Git/GitHub:

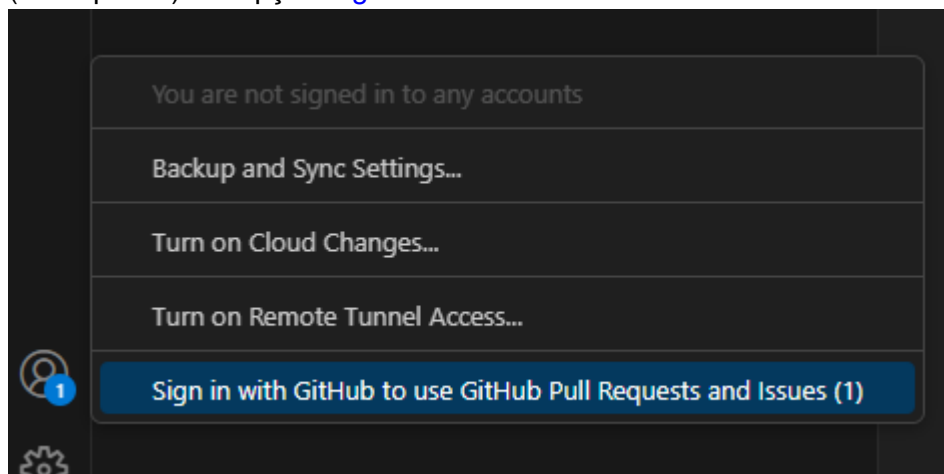
- 1) Crie uma conta, se não tiver, no portal GitHub: <https://github.com/>
  - a) O GitHub é muito utilizado atualmente como forma de apresentar o seu portfólio de desenvolvedor dentro da Computação além de ser útil no desenvolvimento e entrega de trabalhos em equipes.
- 2) Crie um novo repositório para o CCR de Algoritmos no seu GitHub através do botão + no canto superior direito da página do GitHub.



- a) Escolha um nome adequado para o repositório.
- 
- A screenshot of the 'New repository' form on GitHub. The 'Owner' field is set to 'octocat'. The 'Repository name' field contains 'hello-world' and is highlighted with an orange box, with a green checkmark indicating it is valid. Below the name field, there is a suggestion: 'Great repository names are short and memorable. Need inspiration? How about vigilant-meme?'. There is also a 'Description (optional)' field.
- b) Deixe o repositório público e marque para adicionar um README file.
    - c) Mantenha as demais opções nas seleções padrões (em especial, deixe o repositório público para que o professor possa acessar depois para verificar a atividade).
  - 3) Baixe e instale o Git em sua máquina através do sítio: <https://git-scm.com/download>
  - 4) Instale a extensão do [GitHub Pull Requests and Issues](#) no VisualStudio Code
    - a) Revise o Passo 3 da Atividade VSCode para saber como instalar uma extensão no VSCode.
  - 5) Acesse sua conta do GitHub via VSCode através do menu correspondente na aba lateral.
    - a) Siga as instruções de autenticação. Um e-mail com código de verificação será enviado para o seu e-mail cadastrado na conta do GitHub.

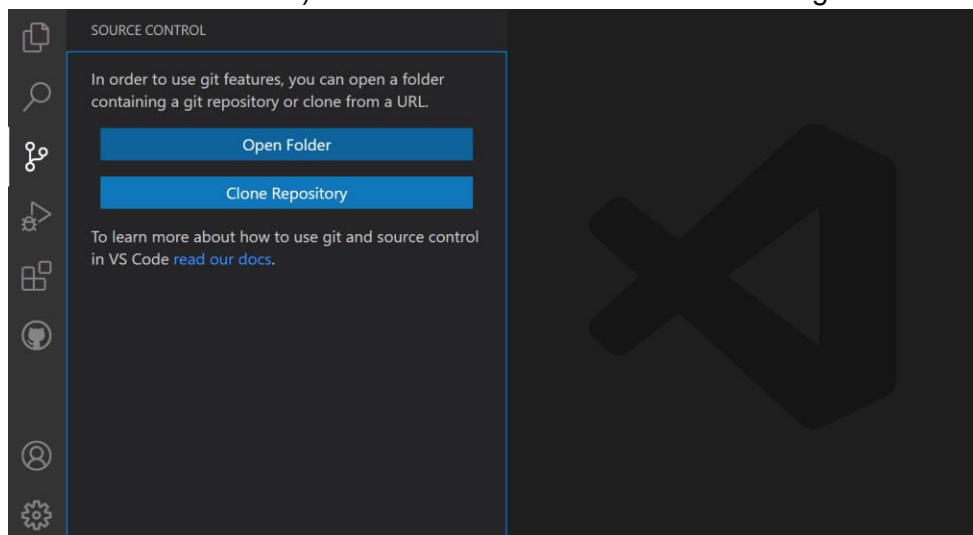


- b) Se a opção da imagem acima não aparecer, acesse o menu [Accounts](#) (bonequinho) e a opção [Sign in...](#)



- 6) Clone o repositório remoto que você criou anteriormente no GitHub. Ao realizar esse processo você estará criando um repositório local com uma cópia do repositório remoto.

- a) Para clonar um repositório acesse o menu de versionamento (terceiro ícone na barra lateral) como mostrado na imagem abaixo.



- b) Selecione a opção clonar do GitHub, aguarde ele listar os projetos e selecione o projeto recém criado.
  - c) Escolha a pasta onde será armazenado o projeto e criado o repositório local.
  - d) Também é possível criar um repositório local primeiro através do Git e depois submetê-lo para o GitHub, mas não será explicado esse processo agora.
- 7) Abra o projeto no VSCode e você verá que o arquivo README aparecerá no VSCode.
- a) Abra o arquivo README e escreva o seu nome e alguma mensagem legal.
  - b) Salve as alterações no arquivo com o atalho Ctrl+S.
- 8) Crie um novo arquivo no projeto chamado de programa.py
- a) Copie um dos programas feitos no CCR de Algoritmos ou crie um novo programinha simples e salve o arquivo.
- 9) Retorne para a aba Source Control do VSCode (imagem do passo 6a) e verá que os dois arquivos aparecerão na lista de mudanças no projeto.
- 10) Digite uma mensagem acima do Commit e pressione o botão.
- a) Você receberá uma mensagem solicitando que as modificações entrem primeiro na área de preparação (stage area). Basta confirmar nesta mensagem.
- 11) Você notará que o botão Commit mudou para Sync Changes. Isso ocorreu pois o seu repositório local está diferente do repositório remoto.
- a) Clique para sincronizar os repositórios para que o VSCode envie suas modificações para o seu repositório remoto.
- 12) Acesse no navegador o seu repositório remoto e confirme que as alterações realizadas localmente foram persistidas no GitHub.
- a) Pode acessar através do GitHub ou do link diretamente. O link para o repositório remoto é algo do tipo:  
<https://github.com/nomeusuariogithub/nomerepositorio>
- 13) **Submeta na atividade do SIGAA** o link com o nome do seu repositório.

Para obter mais informações sobre o uso do Git e GitHub com o VSCode acesse o sítio:

<https://code.visualstudio.com/docs/sourcecontrol/github>