



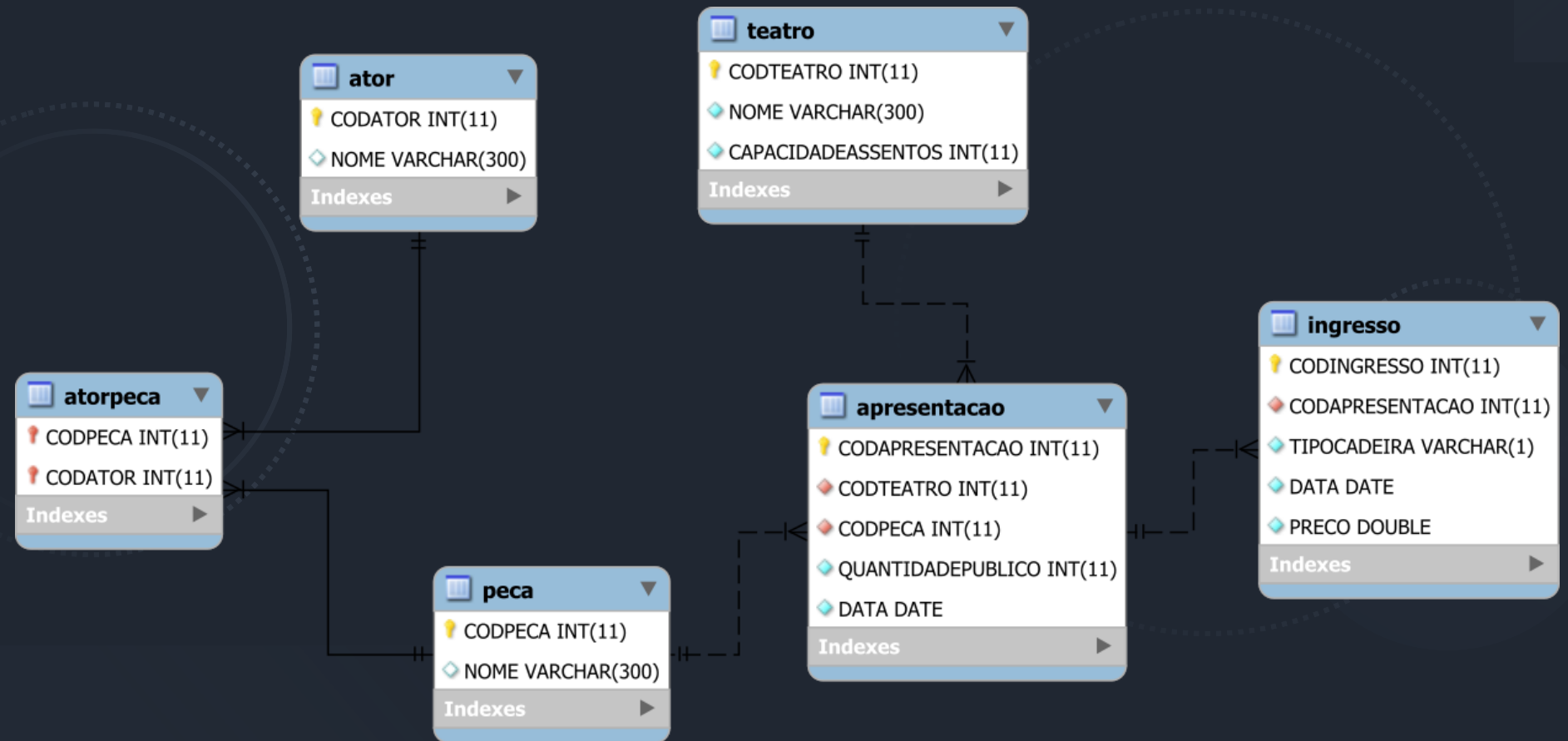
Introdução ao QT Designer com Python

Paulo Vinícius Moreira Dutra¹

¹IF Sudeste MG – Campus Muriaé

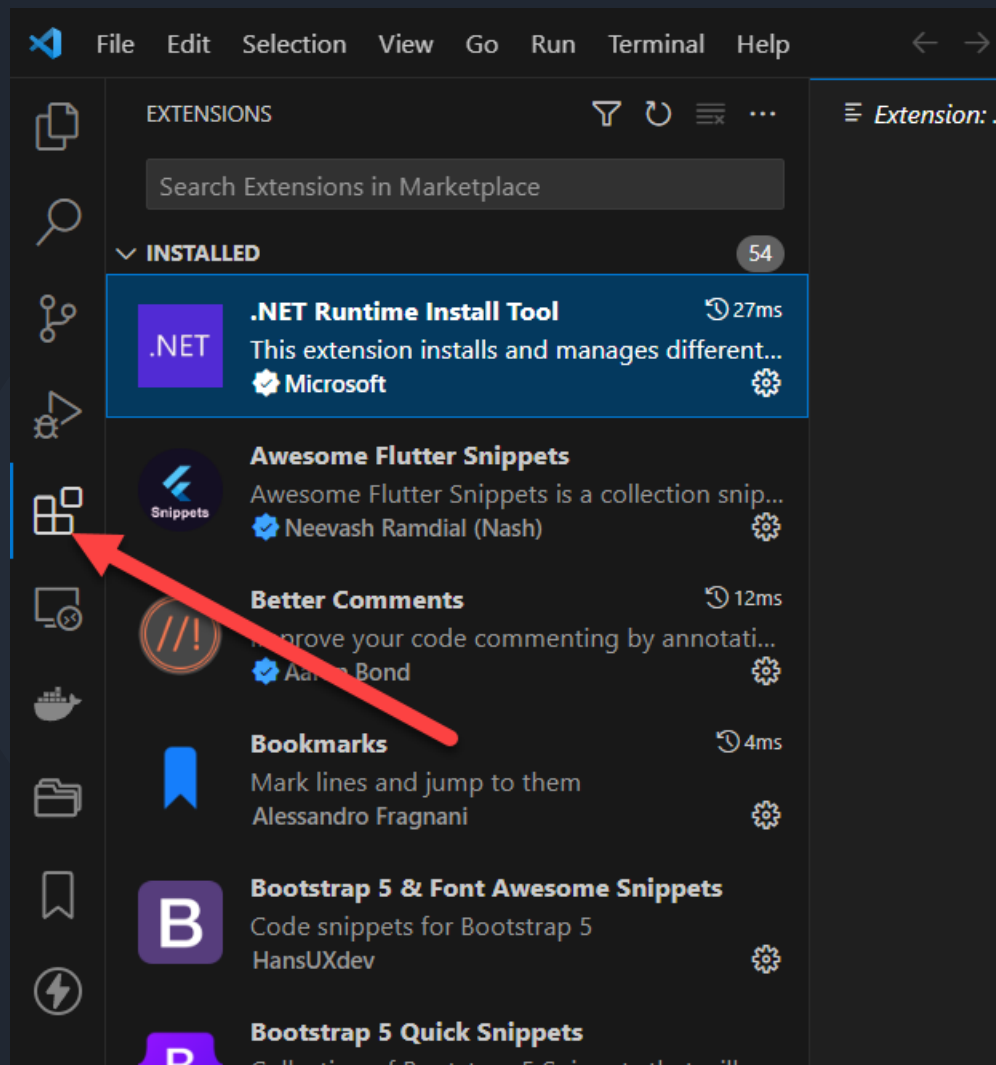
Projeto e Desenvolvimento de Software

Caso de Uso – Sistema para gerenciamento de Teatros



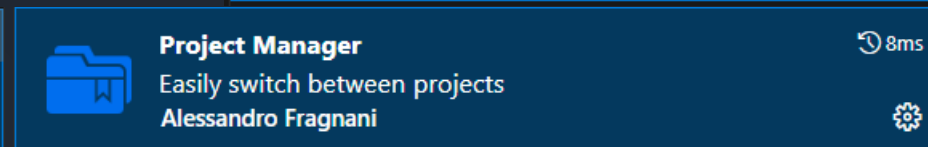
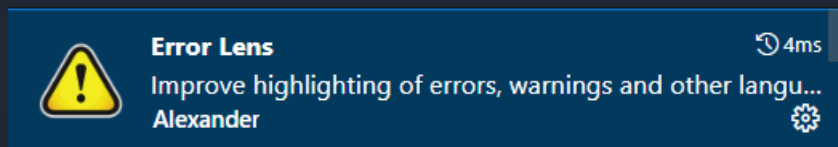
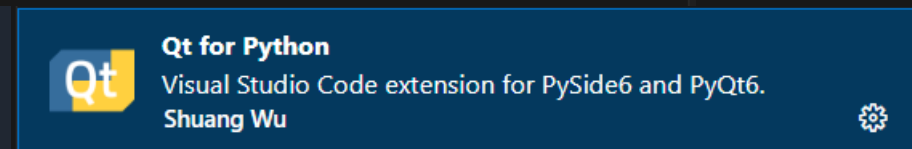
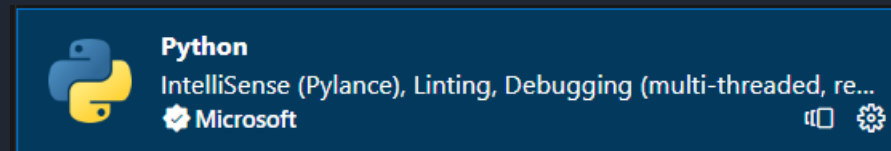
O que você aprenderá

1. Instalar Extensões no Visual Studio Code;
2. Como criar um projeto;
3. Criando um ambiente Virtual em Python;
4. Instalar pacotes no Python;
5. Criar uma conexão com banco de dados utilizando a biblioteca **sqlalchemy**.



1. Instalar Extensões no Visual Studio Code

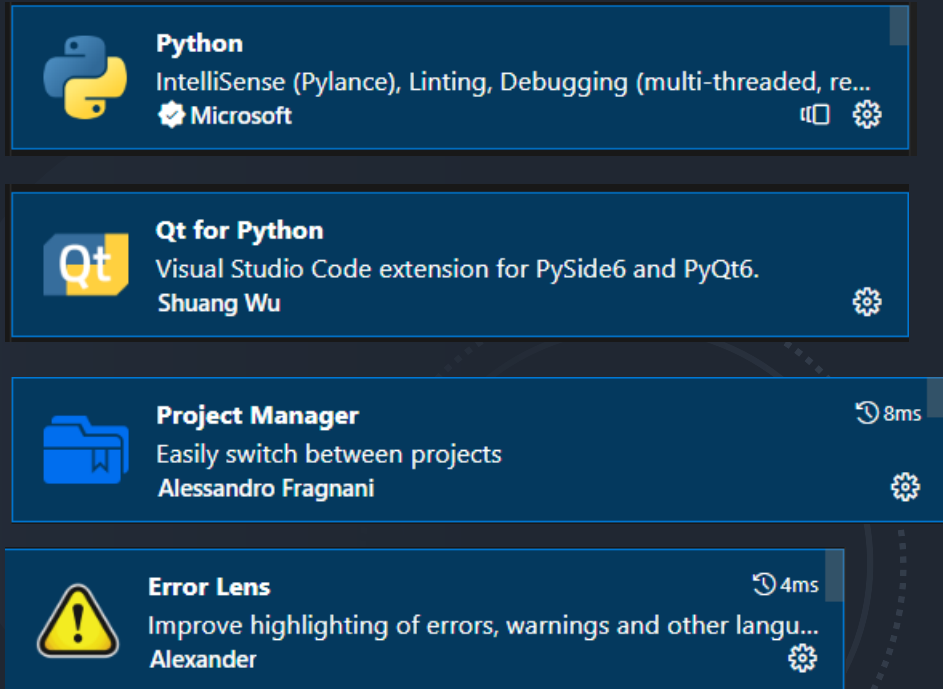
1. Abra o Visual Studio Code;
2. Logo em seguida, clique na opção **Extensions** ou pressione a tecla de atalho **Ctrl + Shift + X**
3. Localize e instale as Extensões **Python**, **Qt for Python**, **Erro Lens** e **Project Manager**.



1. Instalar Extensões no Visual Studio Code

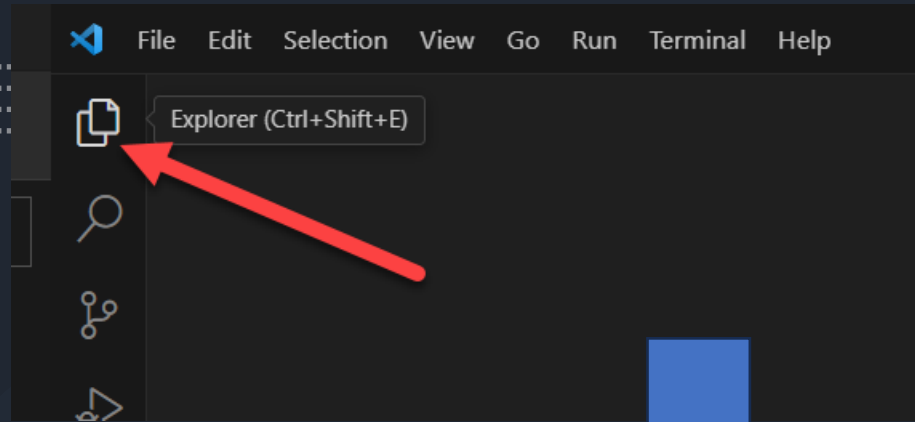
Para um melhor entendimento das extensões, leia o tópico **Extensões de produtividade** da apostila:

https://github.com/dutrapaulovm/PSD/blob/main/Apostila_Did%C3%A1tica_Python_com_sqlalchemy.pdf



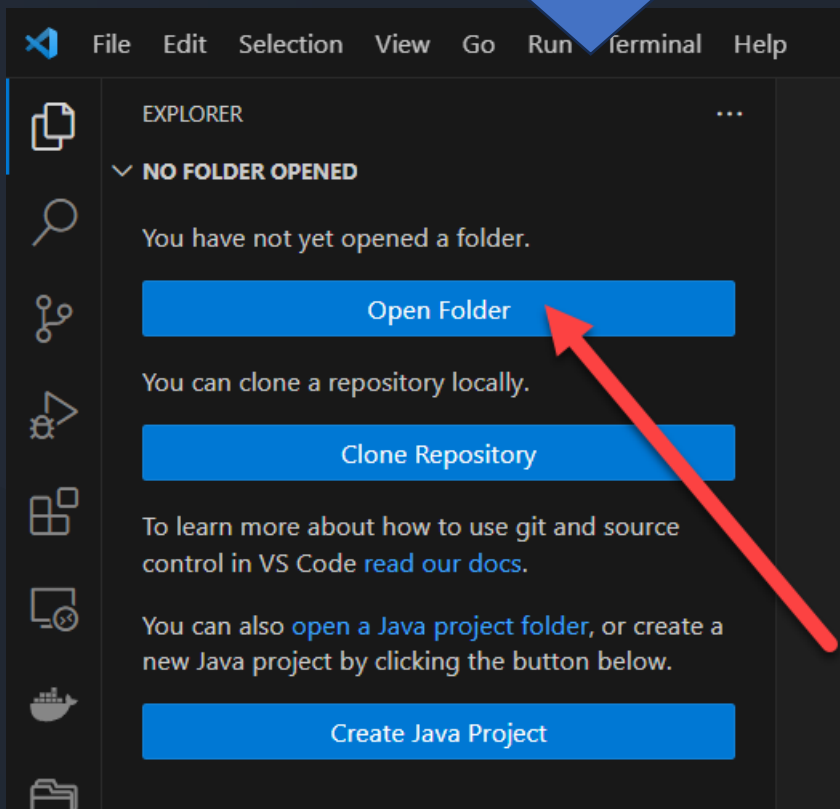
1. Instalar Extensões no Visual Studio Code

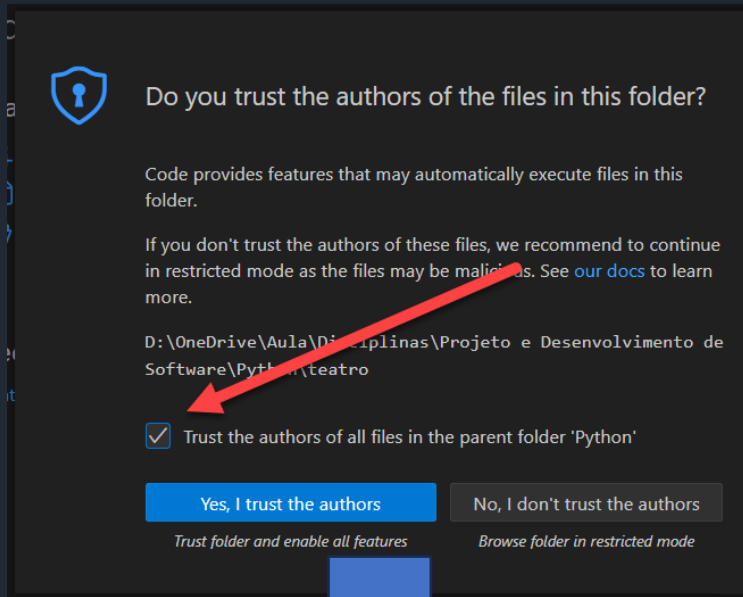




2. Criando um Projeto.

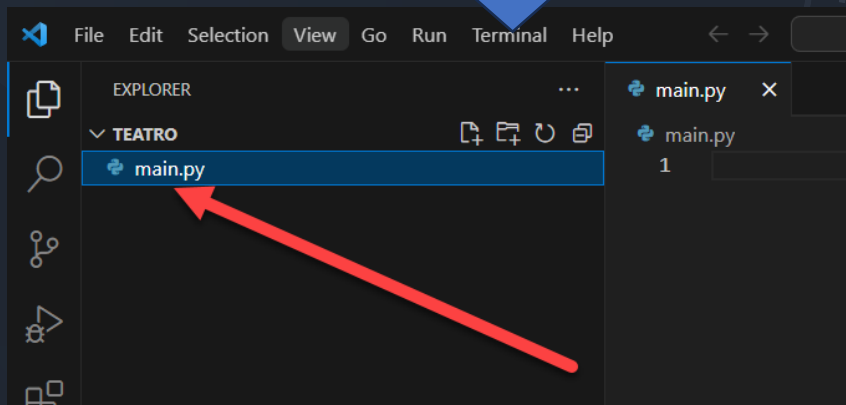
1. Crie uma pasta chamada **teatro** em seu computador no diretório de sua preferência;
2. Abra a pasta teatro no Visual Studio Code utilizando a opção **Explorer** ou pressione a tecla de atalho **Ctrl + Shift + E**.





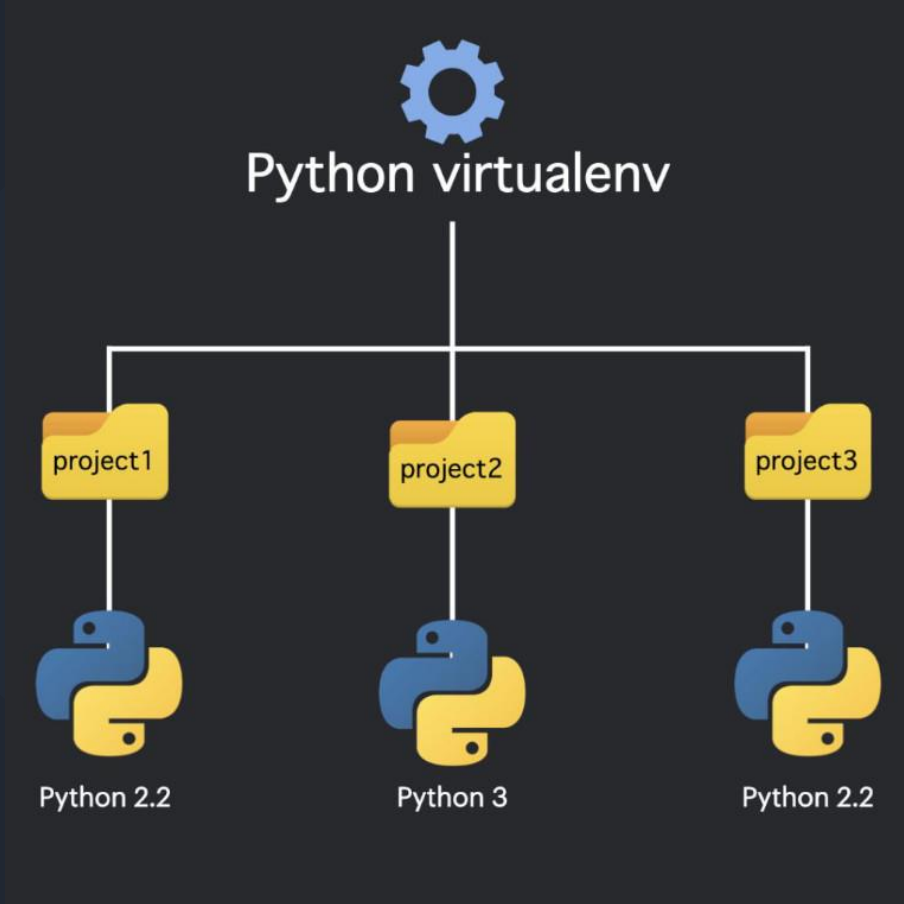
2. Criando um Projeto.

1. Caso apareça a mensagem da Figura ao lado, marque a caixa de seleção e clique no botão **Yes, I trust the authors.**
2. Com a pasta teatro aberta, crie um arquivo chamado **main.py**.
3. O arquivo **main.py** será utilizado para iniciar a nossa aplicação.



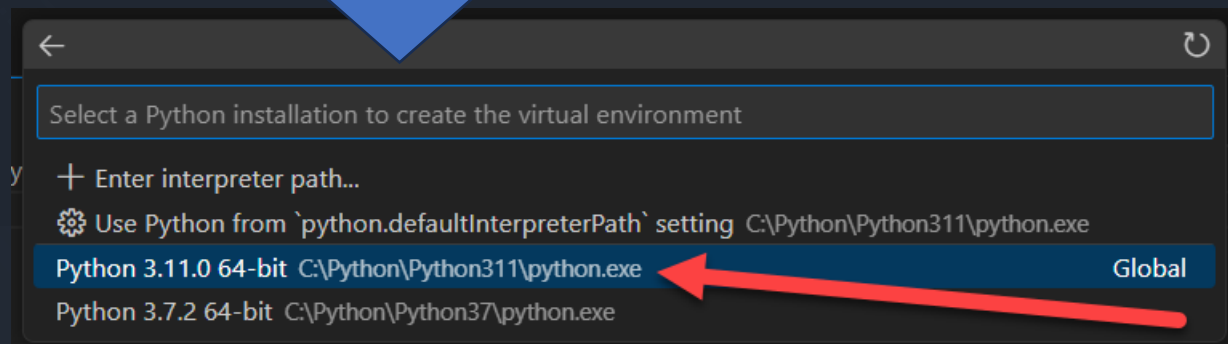
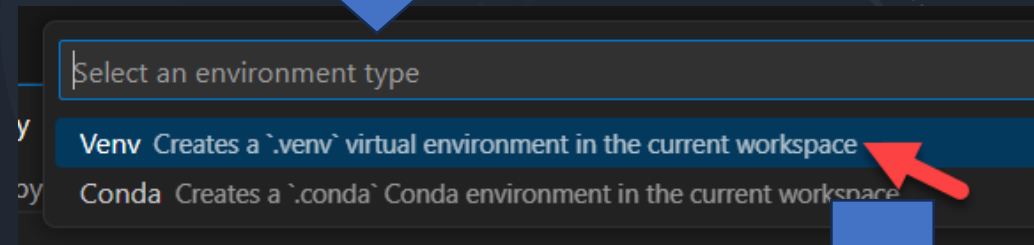
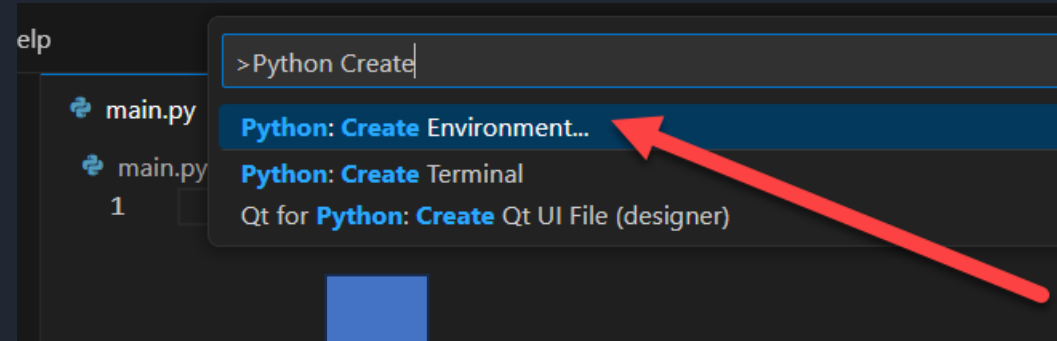
3. Criando um ambiente Virtual em Python

- Um ambiente virtual é uma ferramenta que permite isolar as dependências de um projeto Python.
- Em outras palavras, ele cria um ambiente separado onde você pode instalar bibliotecas e pacotes específicos para o seu projeto sem interferir no ambiente Python global do sistema.
- Para configurar um ambiente virtual em Python no Visual Studio Code utilize a tecla de atalho **Ctrl + Shift + P**. Será exibida uma janela no topo do Visual Studio



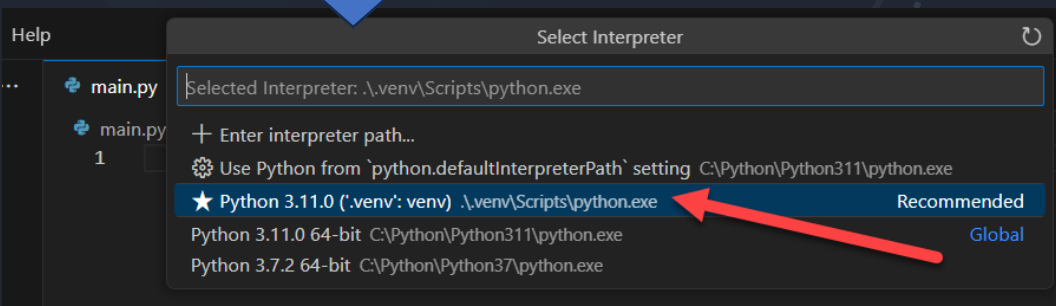
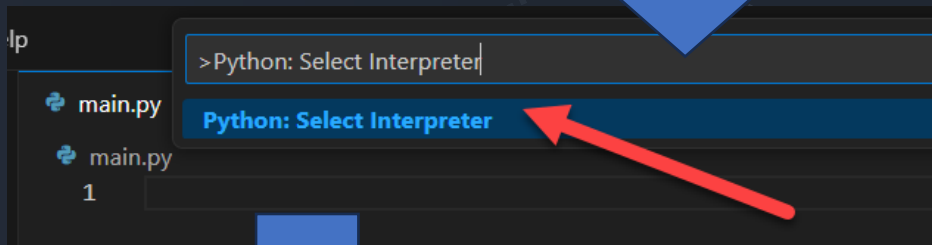
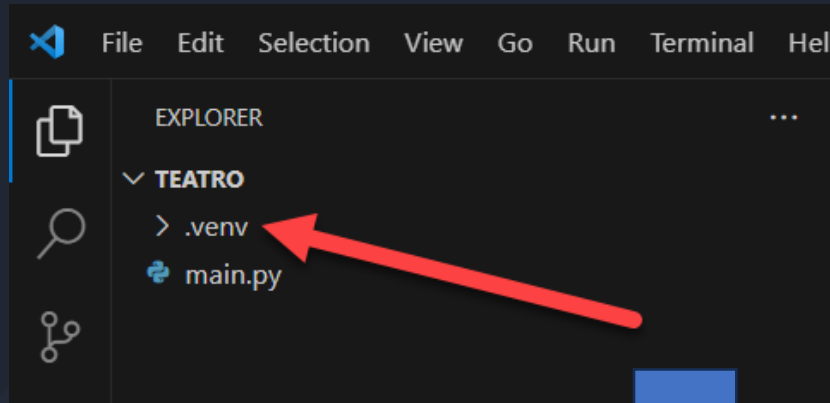
3. Criando um ambiente Virtual em Python

1. Na paleta de comandos, digite "**Python: Create Enviroment**" e pressione Enter.
2. Uma lista suspensa será exibida com as opções de possíveis nomes de ambientes. Selecione o nome **Venv**.
3. Uma lista suspensa será exibida com as opções de interpretadores Python disponíveis. Selecione o interpretador desejado para criar um ambiente virtual



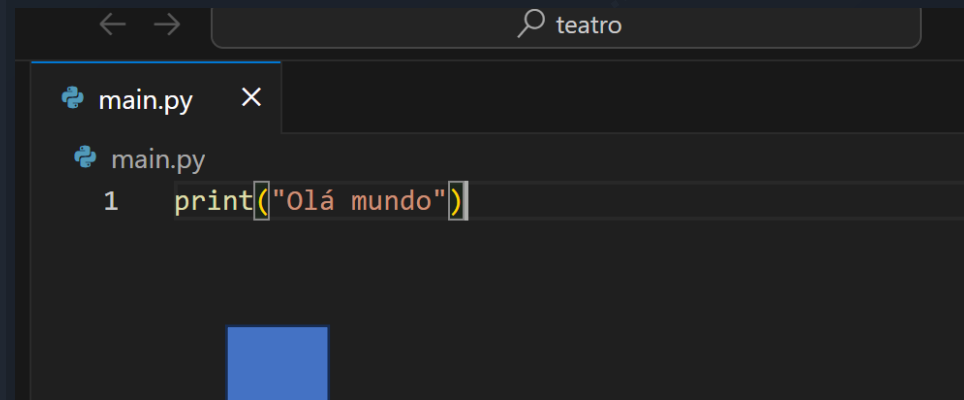
3. Criando um ambiente Virtual em Python

1. Após selecionar o ambiente, será criado uma pasta chamada **.venv** na pasta do projeto **TEATRO**.
2. Agora selecione o ambiente criado pressionando a tecla de atalho **Ctrl + Shift + P** e na paleta de comandos digite "**Python: Select Interpreter**" e pressione Enter.
3. Uma lista suspensa será exibida com as opções de interpretadores Python disponíveis. Selecione o ambiente virtual criado na pasta do projeto.

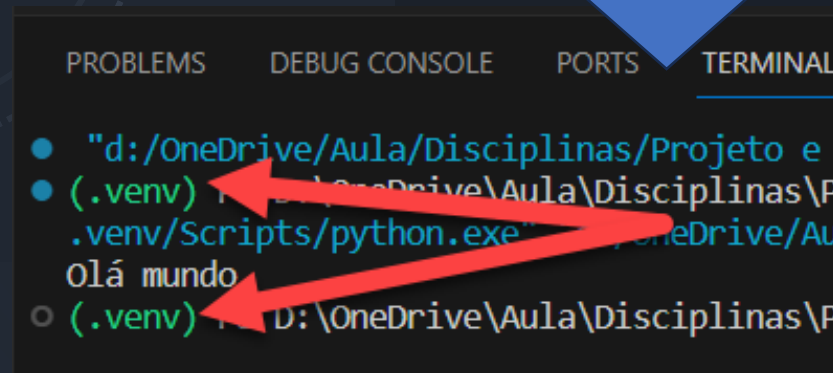


3. Criando um ambiente Virtual em Python

- Agora devemos testar para verificar se o ambiente está sendo chamado corretamente.
- Abra o arquivo **main.py** e digite o código abaixo e compile. No terminal deve aparecer a aplicação executando utilizando o ambiente **(.venv)**



```
main.py  
1 print("Olá mundo")
```



```
PROBLEMS  DEBUG CONSOLE  PORTS  TERMINAL  
• "d:/OneDrive/Aula/Disciplinas/Projeto e  
• (.venv) python.exe "d:/OneDrive/Aula/Disciplinas/P  
  .venv/Scripts/python.exe"  
Olá mundo  
○ (.venv) python.exe "D:\OneDrive\Aula\Disciplinas\P
```

4. Instalando pacotes no Python

1. Os pacotes que iremos instalar a partir desse ponto são essenciais para o correto funcionamento da aplicação em Python.
2. Entretanto, antes de instalar os pacotes, devemos verificar e atualizar o comando **pip**. O comando **pip** é responsável pelas atualizações e instalação de pacotes.
3. Abra o Terminal no Visual Studio Code e instale os seguintes pacotes (Slide seguinte). Para cada comando digitado pressione Enter e aguarde a instalação.

4. Instalando pacotes no Python

1. Abra o Terminal no Visual Studio Code e execute o seguinte comando abaixo:

```
python -m pip install --upgrade pip
```

4. Instalando pacotes no Python

1. Após a atualização do comando **pip**, com o Terminal aberto no Visual Studio Code instale os seguintes pacotes. Para cada comando digitado pressione Enter e aguarde a instalação.

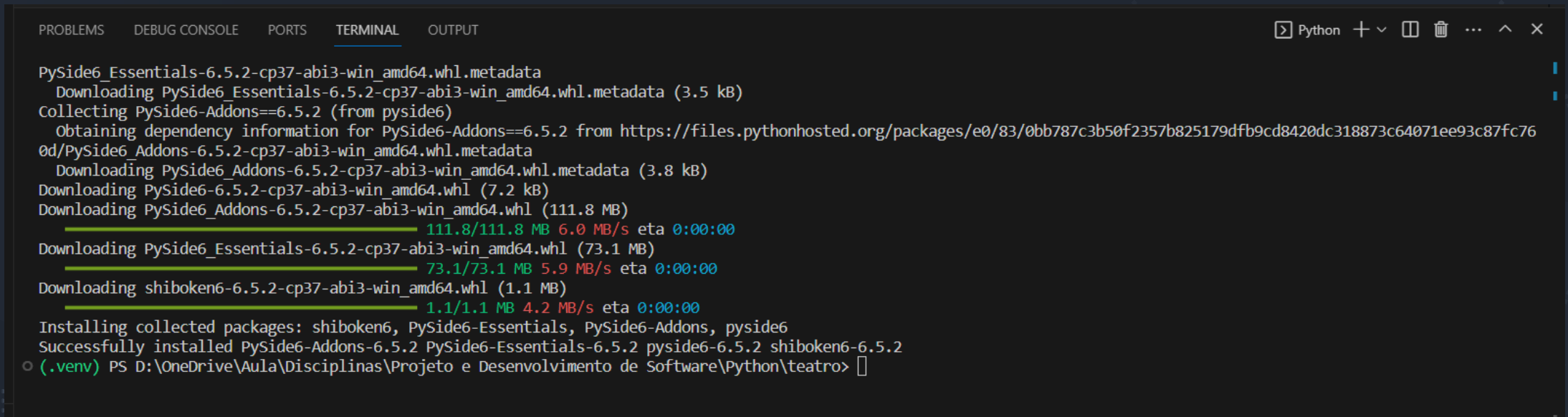
```
pip install sqlalchemy
```

```
pip install pymysql
```

```
pip install pyside6
```

4. Instalando pacotes no Python

- Durante a instalação você verá o progresso ou finalização similar a figura abaixo:



```
PROBLEMS  DEBUG CONSOLE  PORTS  TERMINAL  OUTPUT

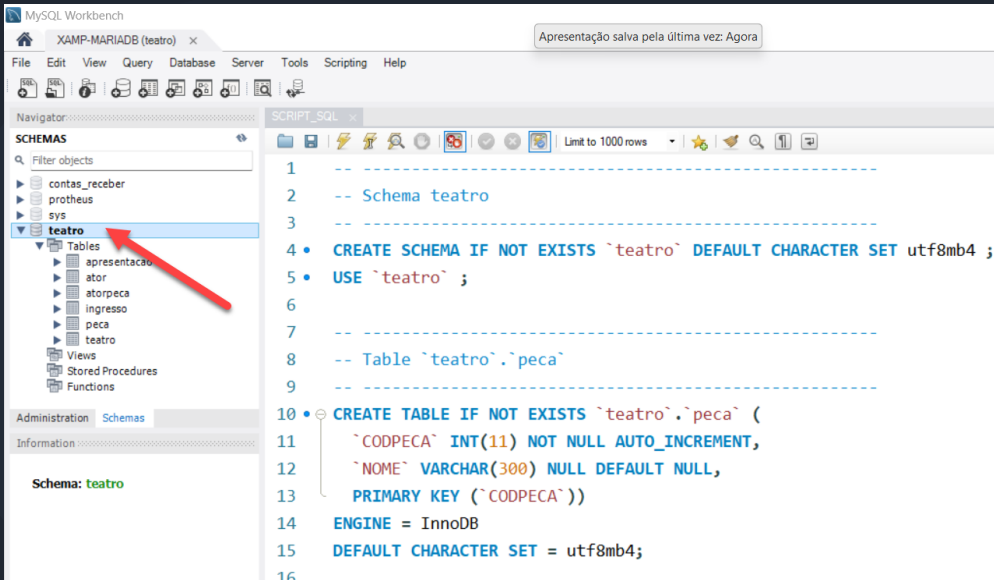
PySide6_Essentials-6.5.2-cp37-abi3-win_amd64.whl.metadata
  Downloading PySide6_Essentials-6.5.2-cp37-abi3-win_amd64.whl.metadata (3.5 kB)
Collecting PySide6-Addons==6.5.2 (from pyside6)
  Obtaining dependency information for PySide6-Addons==6.5.2 from https://files.pythonhosted.org/packages/e0/83/0bb787c3b50f2357b825179dfb9cd8420dc318873c64071ee93c87fc760d/PySide6_Addons-6.5.2-cp37-abi3-win_amd64.whl.metadata
  Downloading PySide6_Addons-6.5.2-cp37-abi3-win_amd64.whl.metadata (3.8 kB)
Downloading PySide6-6.5.2-cp37-abi3-win_amd64.whl (7.2 kB)
Downloading PySide6_Addons-6.5.2-cp37-abi3-win_amd64.whl (111.8 MB)
  111.8/111.8 MB 6.0 MB/s eta 0:00:00
Downloading PySide6_Essentials-6.5.2-cp37-abi3-win_amd64.whl (73.1 MB)
  73.1/73.1 MB 5.9 MB/s eta 0:00:00
Downloading shiboken6-6.5.2-cp37-abi3-win_amd64.whl (1.1 MB)
  1.1/1.1 MB 4.2 MB/s eta 0:00:00
Installing collected packages: shiboken6, PySide6-Essentials, PySide6-Addons, pyside6
Successfully installed PySide6-Addons-6.5.2 PySide6-Essentials-6.5.2 pyside6-6.5.2 shiboken6-6.5.2
(.venv) PS D:\OneDrive\Aula\Disciplinas\Projeto e Desenvolvimento de Software\Python\teatro>
```




5. Criando uma conexão com Banco de dados

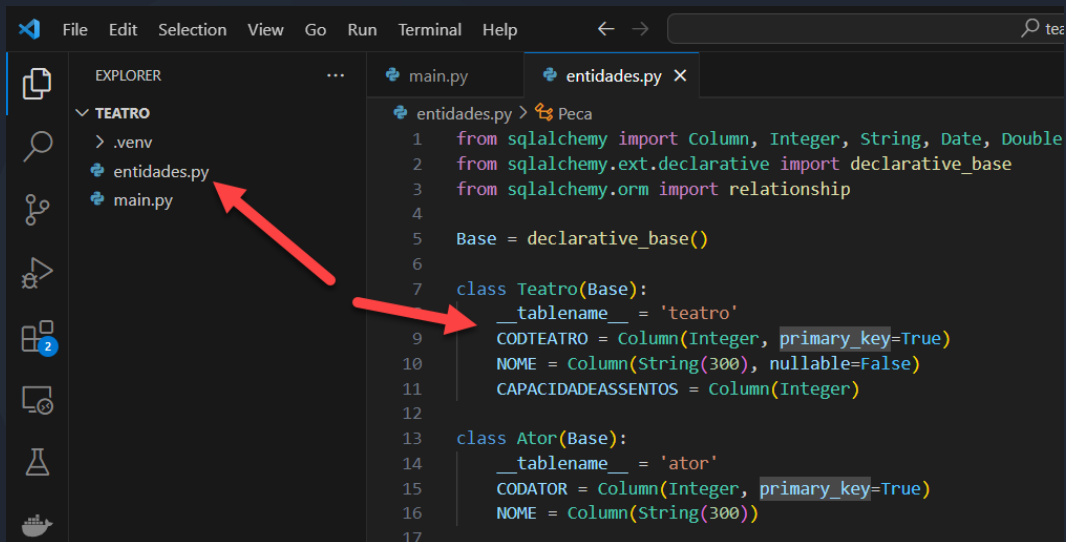
- A partir desse ponto iremos aprender como criar uma conexão com o banco de dados MySQL utilizando a biblioteca **sqlalchemy**.
- Primeiro passo: Criar o banco de dados utilizando o **Workbench**.
- Para o nosso exemplo, iremos utilizar o seguinte Script que pode ser acessado no seguinte link:
https://github.com/dutrapaulovm/PSD/blob/main/SistemaTeatro/SCRIPT_SQL.sql

5. Criando uma conexão com Banco de dados

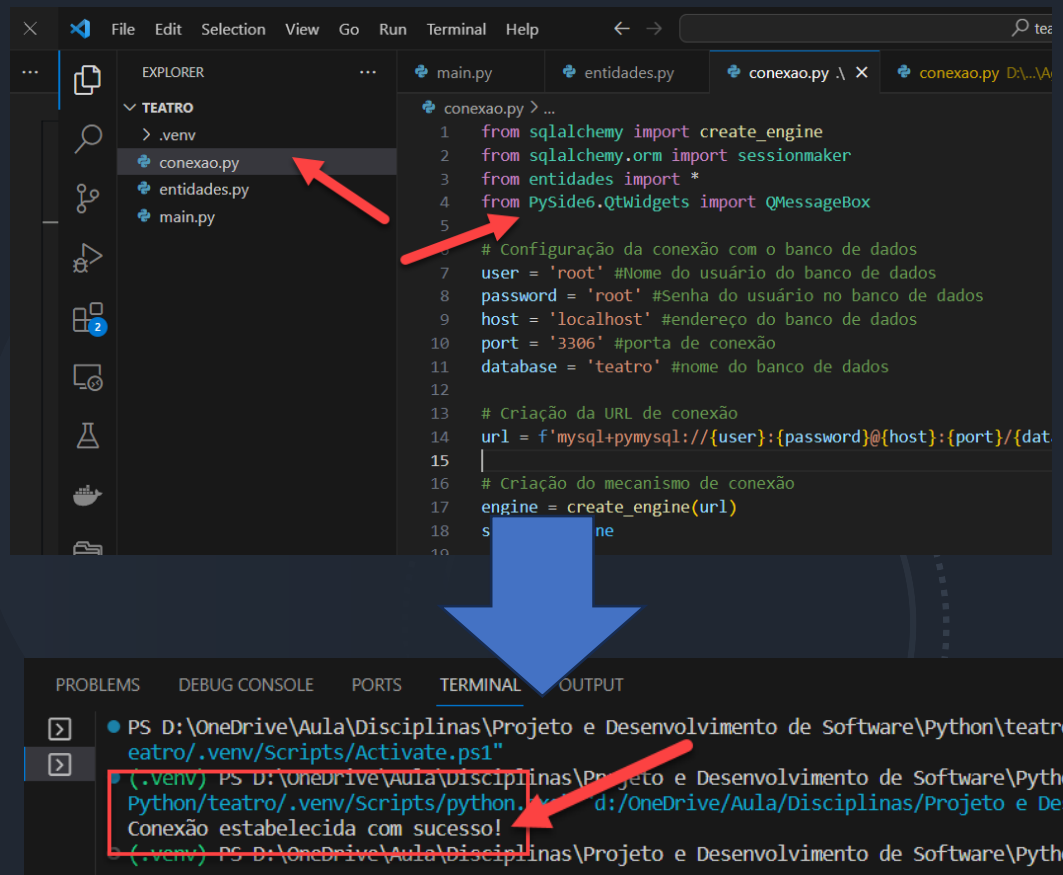


1. Após baixar o Script do banco de dados fornecido no Slide anterior, abra o Workbench, e execute o Script para criar o banco de dados.
2. Agora podemos implementar o código em Python para criar o conexão com o banco de dados.

5. Criando uma conexão com Banco de Dados



1. Com o projeto de **TEATRO** aberto, crie um arquivo chamado entidades.py. No arquivo **entidades.py** será implementado toda a estrutura das entidades do banco de dados. Para cada tabela do banco de dados implementaremos uma classe seguindo o padrão ORM.
2. Baixe o arquivo completo no seguinte link ou se preferir digite o código manualmente:
<https://github.com/dutrapaulovm/PSD/blob/main/teatro/entidades.py>
3. Para a explicação do código, acesse o material da apostila do sqlalchemy. Leia o tópico **Mapeando as tabelas do banco de dados com SQLAlchemy**:
https://github.com/dutrapaulovm/PSD/blob/main/Apostila_Did%C3%A1tica_Python_com_sqlalchemy.pdf



5. Criando uma conexão com Banco de Dados

1. Como próximo passo, vamos criar o arquivo responsável por realizar a conexão com o banco de dados.
2. No projeto **TEATRO** crie um arquivo chamado **conexao.py**. O arquivo completo pode ser baixado no seguinte link:
<https://github.com/dutrapaulovm/PSD/blob/main/teatro/conexao.py>
3. Após implementar o arquivo **conexao.py**, execute e verifique se no Terminal exibiu a mensagem “Conexão estabelecida com sucesso”