

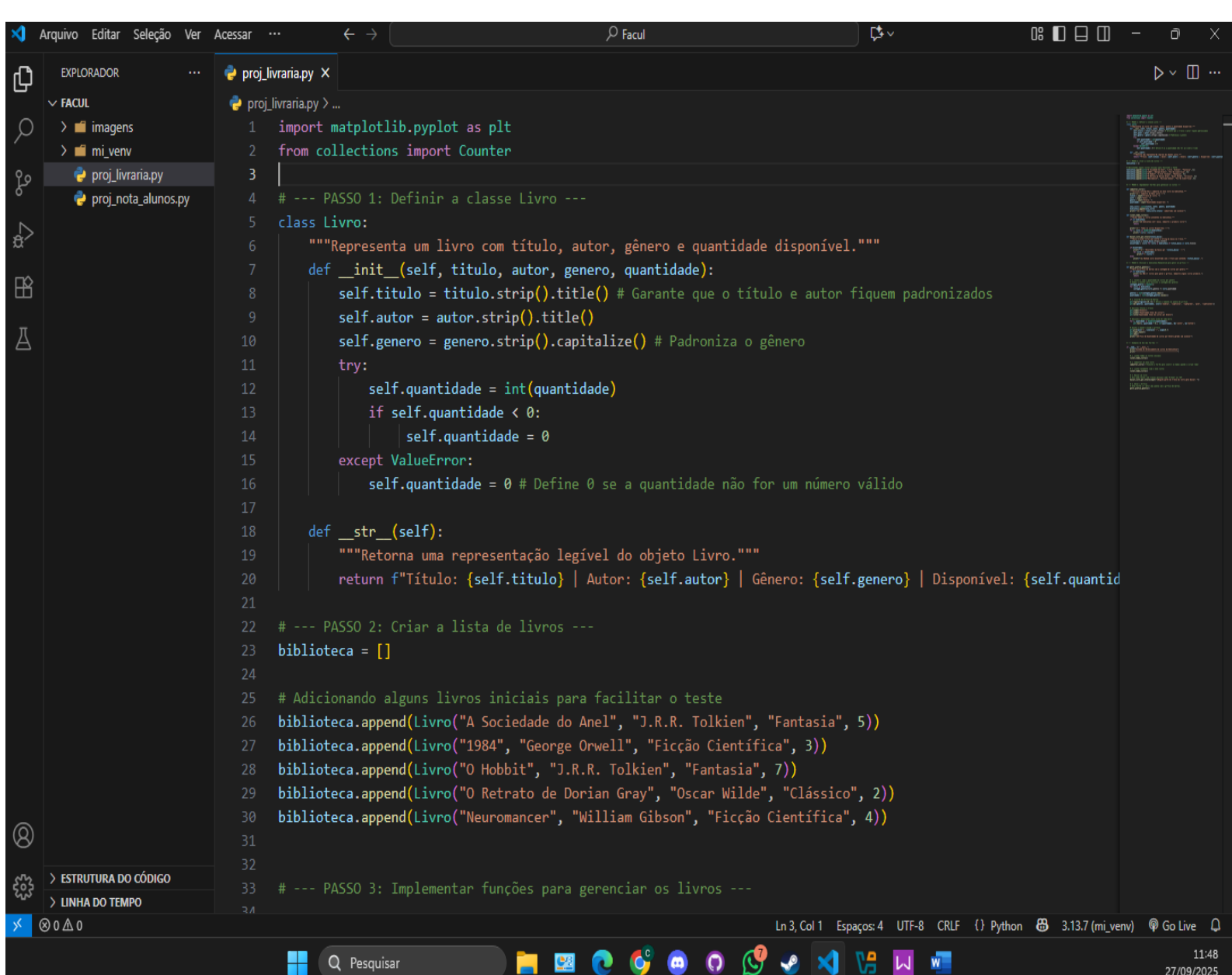
02

TRABALHO DE LINGUAGEM DE PROGRAMAÇÃO EM PYTHON

Estrutura de Dados Central (POO)

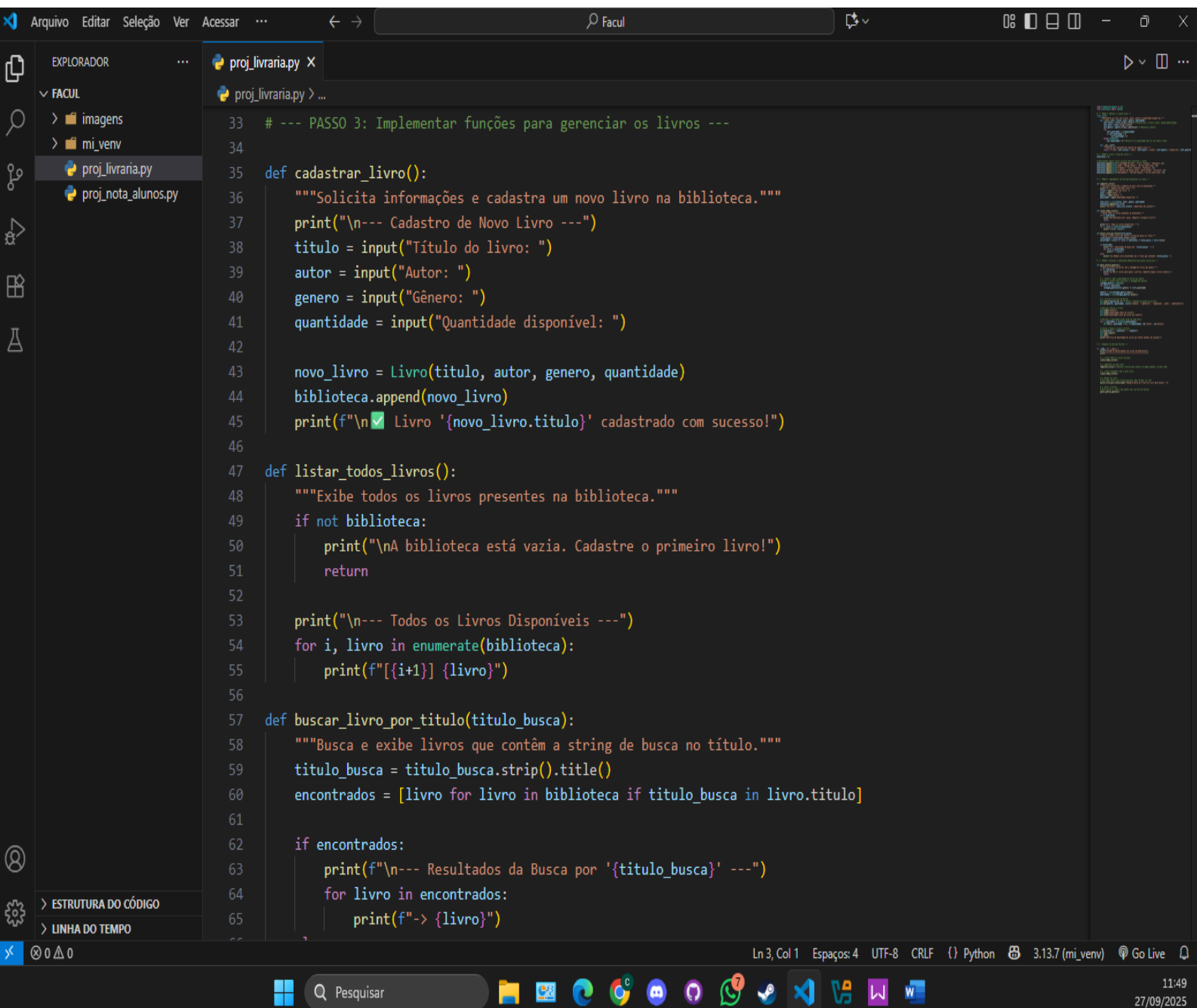
O projeto é construído em torno da classe **Livro**, que é o seu modelo de dados fundamental.

- **Classe Livro:** Define como um livro é estruturado no sistema, encapsulando seus atributos (**título, autor, gênero, quantidade**).



```
1 import matplotlib.pyplot as plt
2 from collections import Counter
3
4 # --- PASSO 1: Definir a classe Livro ---
5 class Livro:
6     """Representa um livro com título, autor, gênero e quantidade disponível."""
7     def __init__(self, titulo, autor, genero, quantidade):
8         self.titulo = titulo.strip().title() # Garante que o título e autor fiquem padronizados
9         self.autor = autor.strip().title()
10        self.genero = genero.strip().capitalize() # Padroniza o gênero
11        try:
12            self.quantidade = int(quantidade)
13            if self.quantidade < 0:
14                self.quantidade = 0
15        except ValueError:
16            self.quantidade = 0 # Define 0 se a quantidade não for um número válido
17
18        def __str__(self):
19            """Retorna uma representação legível do objeto livro."""
20            return f"Título: {self.titulo} | Autor: {self.autor} | Gênero: {self.genero} | Disponível: {self.quantidade}"
21
22 # --- PASSO 2: Criar a lista de livros ---
23 biblioteca = []
24
25 # Adicionando alguns livros iniciais para facilitar o teste
26 biblioteca.append(Livro("A Sociedade do Anel", "J.R.R. Tolkien", "Fantasia", 5))
27 biblioteca.append(Livro("1984", "George Orwell", "Ficção Científica", 3))
28 biblioteca.append(Livro("O Hobbit", "J.R.R. Tolkien", "Fantasia", 7))
29 biblioteca.append(Livro("O Retrato de Dorian Gray", "Oscar Wilde", "Clássico", 2))
30 biblioteca.append(Livro("Neuromancer", "William Gibson", "Ficção Científica", 4))
31
32
33 # --- PASSO 3: Implementar funções para gerenciar os livros ---
34
```

As funções de gerenciamento (**CRUD - Create, Read, Update, Delete**) são implementadas como módulos separados, interagindo com o **catalogo livros**.

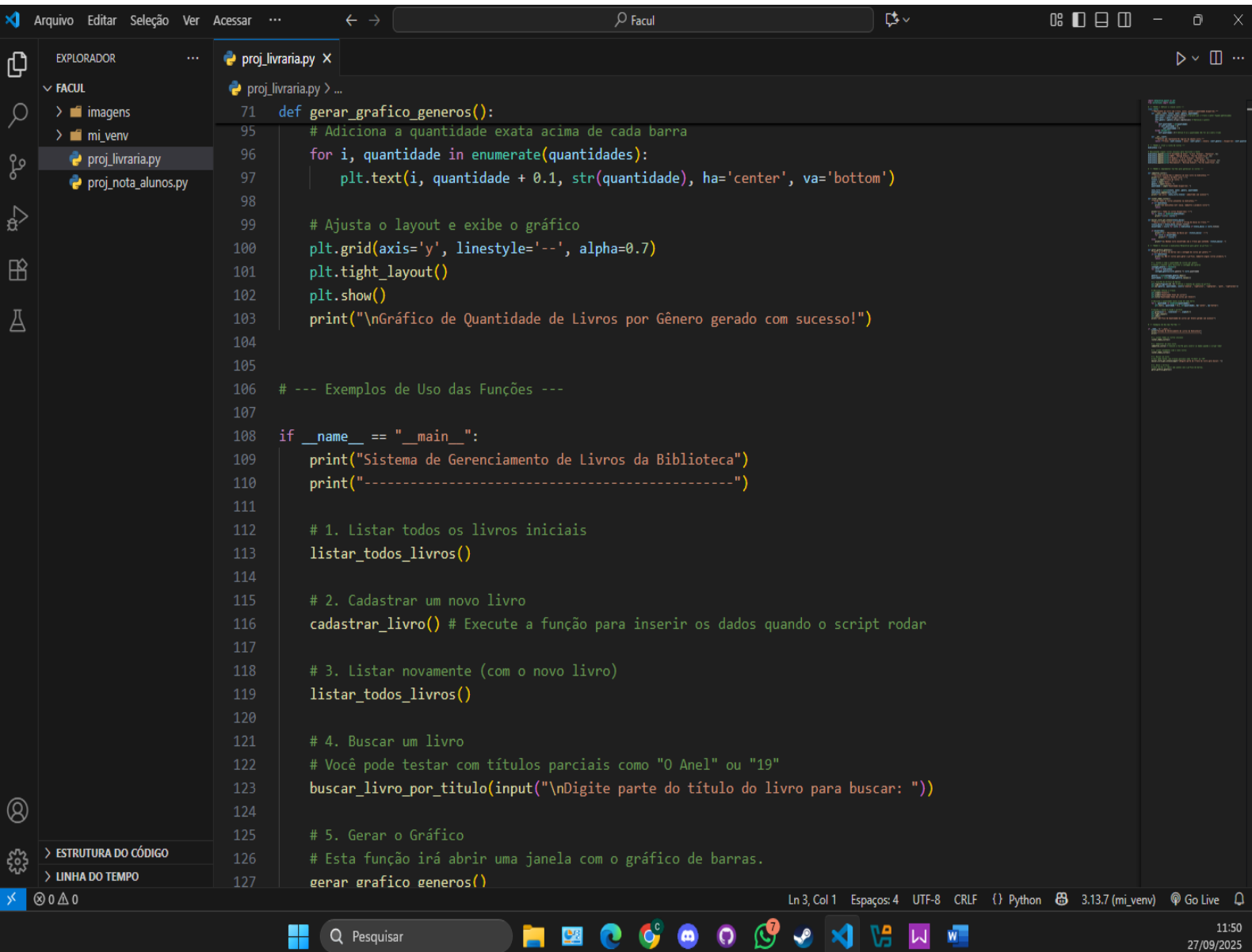


```
33 # --- PASSO 3: Implementar funções para gerenciar os livros ---
34
35 def cadastrar_livro():
36     """Solicita informações e cadastra um novo livro na biblioteca."""
37     print("\n--- Cadastro de Novo Livro ---")
38     titulo = input("Título do livro: ")
39     autor = input("Autor: ")
40     genero = input("Gênero: ")
41     quantidade = input("Quantidade disponível: ")
42
43     novo_livro = Livro(titulo, autor, genero, quantidade)
44     biblioteca.append(novo_livro)
45     print(f"\n✅ Livro '{novo_livro.titulo}' cadastrado com sucesso!")
46
47 def listar_todos_livros():
48     """Exibe todos os livros presentes na biblioteca."""
49     if not biblioteca:
50         print("\nA biblioteca está vazia. Cadastre o primeiro livro!")
51         return
52
53     print("\n--- Todos os Livros Disponíveis ---")
54     for i, livro in enumerate(biblioteca):
55         print(f"[{i+1}] {livro}")
56
57 def buscar_livro_por_titulo(titulo_busca):
58     """Busca e exibe livros que contêm a string de busca no título."""
59     titulo_busca = titulo_busca.strip().title()
60     encontrados = [livro for livro in biblioteca if titulo_busca in livro.titulo]
61
62     if encontrados:
63         print(f"\n--- Resultados da Busca por '{titulo_busca}' ---")
64         for livro in encontrados:
65             print(f"-> {livro}")
```

Módulo de Visualização de Dados

A função **gerar_grafico_generos()** é responsável por analisar os dados brutos e transformá-los em informação visual.

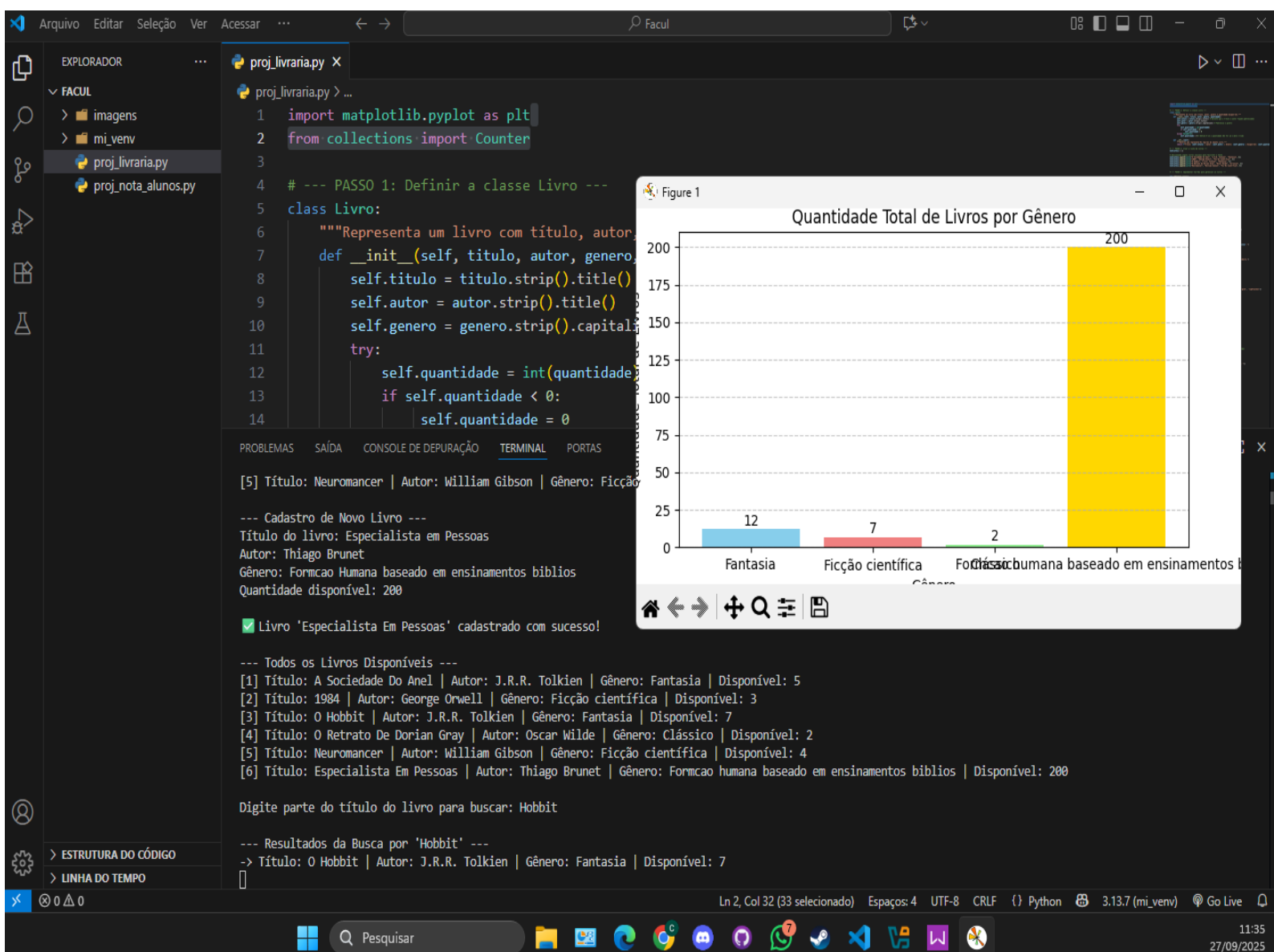
- **Coleta de Dados:** Utiliza uma *list comprehension* para extrair a lista de **todos os gêneros** presentes no catálogo.
- **Contagem:** Emprega a classe **Counter** do módulo `collections` para contar a frequência de cada gênero, gerando um dicionário de {Gênero: Quantidade}. Esta é a etapa lógica crucial para sumarizar os dados.



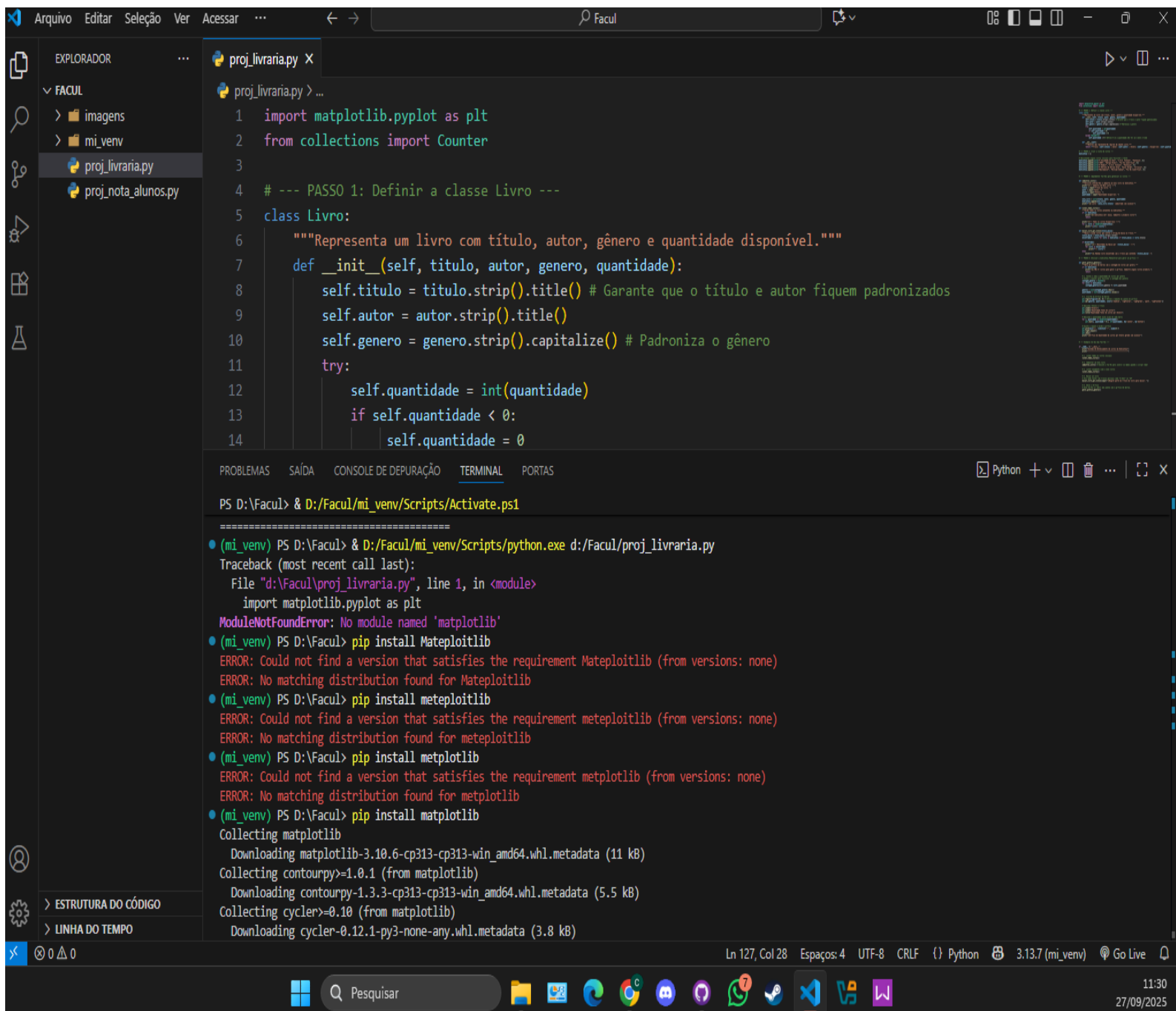
```
71 def gerar_grafico_generos():
72     # Adiciona a quantidade exata acima de cada barra
73     for i, quantidade in enumerate(quantidades):
74         plt.text(i, quantidade + 0.1, str(quantidade), ha='center', va='bottom')
75
76     # Ajusta o layout e exibe o gráfico
77     plt.grid(axis='y', linestyle='--', alpha=0.7)
78     plt.tight_layout()
79     plt.show()
80     print("\nGráfico de Quantidade de Livros por Gênero gerado com sucesso!")
81
82 # --- Exemplos de Uso das Funções ---
83
84 if __name__ == "__main__":
85     print("Sistema de Gerenciamento de Livros da Biblioteca")
86     print("-----")
87
88     # 1. Listar todos os livros iniciais
89     listar_todos_livros()
90
91     # 2. Cadastrar um novo livro
92     cadastrar_livro() # Execute a função para inserir os dados quando o script rodar
93
94     # 3. Listar novamente (com o novo livro)
95     listar_todos_livros()
96
97     # 4. Buscar um livro
98     # Você pode testar com títulos parciais como "O Anel" ou "19"
99     buscar_livro_por_titulo(input("\nDigite parte do título do livro para buscar: "))
100
101     # 5. Gerar o Gráfico
102     # Esta função irá abrir uma janela com o gráfico de barras.
103     gerar_grafico_generos()
```

Visualização: Utiliza o **matplotlib.pyplot** para:

- Criar um gráfico de barras (plt.bar).
- Definir rótulos e títulos.
- Adicionar rótulos numéricos diretamente sobre as barras, o que aumenta a legibilidade.
- Rotacionar os rótulos do eixo X (plt.xticks(rotation=45)) para melhor manuseio de nomes de gêneros longos.



Instalação do METPLOTLIB:



The image shows a Visual Studio Code editor window with a Python file named `proj_livraria.py` open. The file contains a class `Livro` with attributes `titulo`, `autor`, `genero`, and `quantidade`. The terminal window at the bottom shows the following commands and output:

```
PS D:\Facul> & D:/Facul/mi_env/Scripts/Activate.ps1

=====
(mi_env) PS D:\Facul> & D:/Facul/mi_env/Scripts/python.exe d:/Facul/proj_livraria.py
Traceback (most recent call last):
  File "d:\Facul\proj_livraria.py", line 1, in <module>
    import matplotlib.pyplot as plt
ModuleNotFoundError: No module named 'matplotlib'

(mi_env) PS D:\Facul> pip install Mateploitleib
ERROR: Could not find a version that satisfies the requirement Mateploitleib (from versions: none)
ERROR: No matching distribution found for Mateploitleib

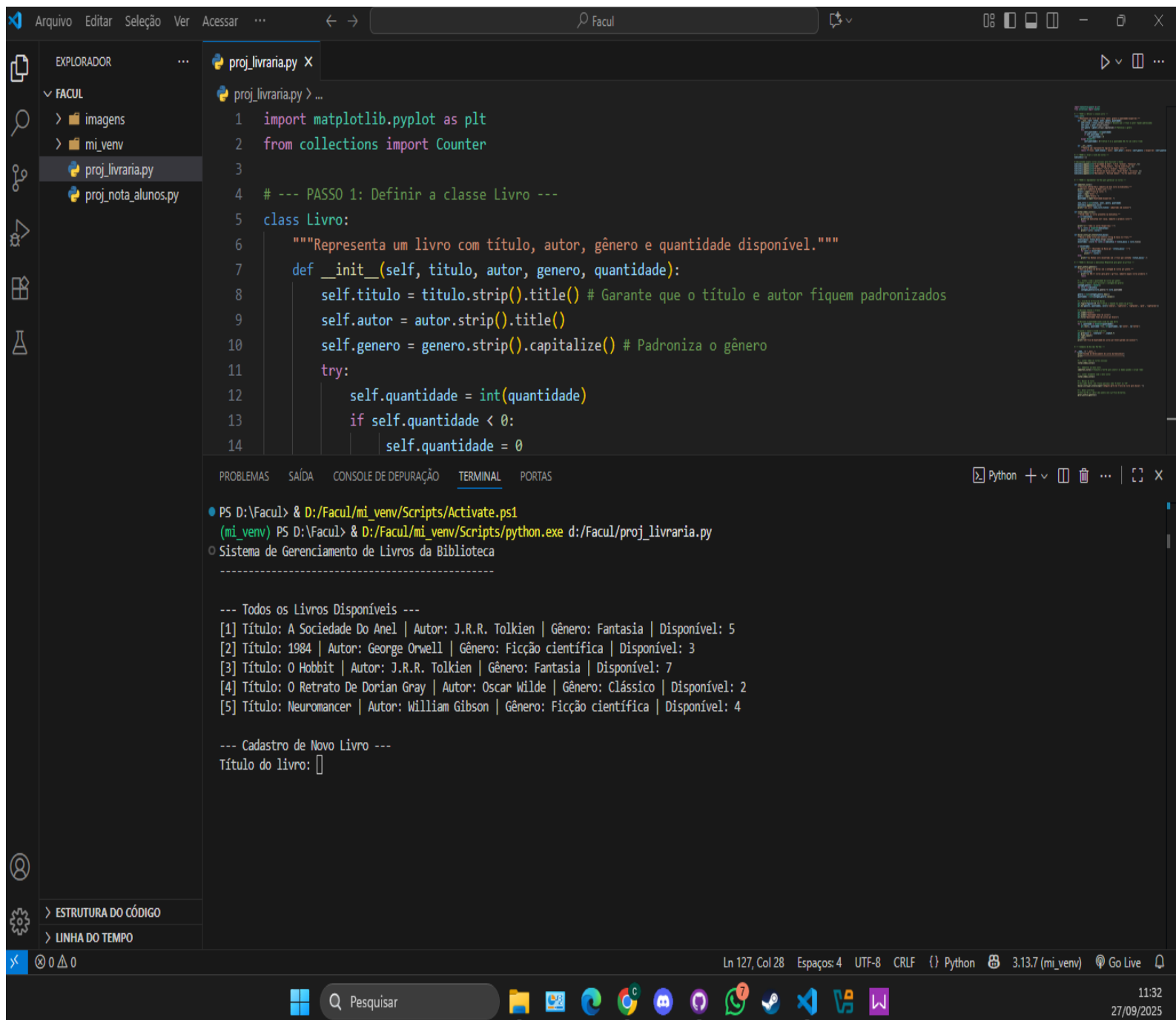
(mi_env) PS D:\Facul> pip install meteploitleib
ERROR: Could not find a version that satisfies the requirement meteploitleib (from versions: none)
ERROR: No matching distribution found for meteploitleib

(mi_env) PS D:\Facul> pip install metplotlib
ERROR: Could not find a version that satisfies the requirement metplotlib (from versions: none)
ERROR: No matching distribution found for metplotlib

(mi_env) PS D:\Facul> pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.10.6-cp313-cp313-win_amd64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.3-cp313-cp313-win_amd64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
```

The status bar at the bottom indicates the current file is at line 127, column 28, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.13.7 (mi_env), and Go Live mode.

Primeira Função com o Menu aparente:



The image shows a Visual Studio Code editor window with a Python file named `proj_livraria.py` open. The file contains a class `Livro` and a list of books. The terminal output shows the execution of the script, displaying a list of books and a prompt for a new book.

```
1 import matplotlib.pyplot as plt
2 from collections import Counter
3
4 # --- PASSO 1: Definir a classe Livro ---
5 class Livro:
6     """Representa um livro com título, autor, gênero e quantidade disponível."""
7     def __init__(self, titulo, autor, genero, quantidade):
8         self.titulo = titulo.strip().title() # Garante que o título e autor fiquem padronizados
9         self.autor = autor.strip().title()
10        self.genero = genero.strip().capitalize() # Padroniza o gênero
11        try:
12            self.quantidade = int(quantidade)
13            if self.quantidade < 0:
14                self.quantidade = 0
```

Terminal Output:

```
PS D:\Facul> & D:/Facul/mi_venv/Scripts/Activate.ps1
(mi_venv) PS D:\Facul> & D:/Facul/mi_venv/Scripts/python.exe d:/Facul/proj_livraria.py
Sistema de Gerenciamento de Livros da Biblioteca

--- Todos os Livros Disponíveis ---
[1] Título: A Sociedade Do Anel | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 5
[2] Título: 1984 | Autor: George Orwell | Gênero: Ficção científica | Disponível: 3
[3] Título: O Hobbit | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 7
[4] Título: O Retrato De Dorian Gray | Autor: Oscar Wilde | Gênero: Clássico | Disponível: 2
[5] Título: Neuromancer | Autor: William Gibson | Gênero: Ficção científica | Disponível: 4

--- Cadastro de Novo Livro ---
Título do livro: 
```

Cadastro de um novo Livro:

The image shows a Windows 11 desktop with a Visual Studio Code (VS Code) editor open. The editor is displaying a Python file named 'proj_livraria.py'. The code defines a 'Livro' class with attributes for title, author, genre, and quantity, and a main function that interacts with a database to list and add books. The terminal window at the bottom shows the output of the program, including a list of books and a confirmation message for adding a new book. The file explorer on the left shows the project structure, including folders 'imagens' and 'mi_venv', and files 'proj_livraria.py' and 'proj_nota_alunos.py'. The taskbar at the bottom shows various application icons, including the Start menu, search bar, and several open applications like File Explorer, Edge, and VS Code. The system clock in the bottom right corner shows the date and time as 11:34 on 27/09/2025.

Função -> Buscar por partes do **Título do Livro:**

The image shows a VS Code editor window with a Python file named 'proj_livraria.py'. The code defines a 'Livro' class with attributes for title, author, genre, and quantity. It includes a main function that lists books, registers a new book, and lists all books. The terminal output shows the execution results, including a list of books and a confirmation message for a new book registration.

```
1 import matplotlib.pyplot as plt
2 from collections import Counter
3
4 # --- PASSO 1: Definir a classe Livro ---
5 class Livro:
6     """Representa um livro com título, autor, gênero e quantidade disponível."""
7     def __init__(self, titulo, autor, genero, quantidade):
8         self.titulo = titulo.strip().title() # Garante que o título e autor fiquem padronizados
9         self.autor = autor.strip().title()
10        self.genero = genero.strip().capitalize() # Padroniza o gênero
11        try:
12            self.quantidade = int(quantidade)
13            if self.quantidade < 0:
14                self.quantidade = 0
```

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO TERMINAL PORTAS

[1] Título: A Sociedade Do Anel | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 5
[2] Título: 1984 | Autor: George Orwell | Gênero: Ficção científica | Disponível: 3
[3] Título: O Hobbit | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 7
[4] Título: O Retrato De Dorian Gray | Autor: Oscar Wilde | Gênero: Clássico | Disponível: 2
[5] Título: Neuromancer | Autor: William Gibson | Gênero: Ficção científica | Disponível: 4

--- Cadastro de Novo Livro ---
Título do livro: Especialista em Pessoas
Autor: Thiago Brunet
Gênero: Formcao Humana baseado em ensinamentos biblicos
Quantidade disponível: 200

✔ Livro 'Especialista Em Pessoas' cadastrado com sucesso!

--- Todos os Livros Disponíveis ---
[1] Título: A Sociedade Do Anel | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 5
[2] Título: 1984 | Autor: George Orwell | Gênero: Ficção científica | Disponível: 3
[3] Título: O Hobbit | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 7
[4] Título: O Retrato De Dorian Gray | Autor: Oscar Wilde | Gênero: Clássico | Disponível: 2
[5] Título: Neuromancer | Autor: William Gibson | Gênero: Ficção científica | Disponível: 4
[6] Título: Especialista Em Pessoas | Autor: Thiago Brunet | Gênero: Formcao humana baseado em ensinamentos biblicos | Disponível: 200

Digite parte do título do livro para buscar:

Mensagem de Sucesso pelo Gráfico Gerado:

The image shows a Visual Studio Code editor window with a Python file named `proj_livraria.py` open. The code defines a `Livro` class and uses `matplotlib` and `collections.Counter`. The terminal output shows the successful execution of the script, displaying a list of books and their quantities, and a search result for 'Hobbit'.

```
1 import matplotlib.pyplot as plt
2 from collections import Counter
3
4 # --- PASSO 1: Definir a classe Livro ---
5 class Livro:
6     """Representa um livro com título, autor, gênero e quantidade disponível."""
7     def __init__(self, titulo, autor, genero, quantidade):
8         self.titulo = titulo.strip().title() # Garante que o título e autor fiquem padronizados
9         self.autor = autor.strip().title()
10        self.genero = genero.strip().capitalize() # Padroniza o gênero
11        try:
12            self.quantidade = int(quantidade)
13        except:
14            self.quantidade = 0
```

Terminal Output:

```
Sistema de Gerenciamento de Livros da Biblioteca

Título do livro: Especialista em Pessoas
Autor: Thiago Brunet
Gênero: Formcao Humana baseado em ensinamentos biblios
Quantidade disponível: 200

✅ Livro 'Especialista Em Pessoas' cadastrado com sucesso!

--- Todos os Livros Disponíveis ---
[1] Título: A Sociedade Do Anel | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 5
[2] Título: 1984 | Autor: George Orwell | Gênero: Ficção científica | Disponível: 3
[3] Título: O Hobbit | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 7
[4] Título: O Retrato De Dorian Gray | Autor: Oscar Wilde | Gênero: Clássico | Disponível: 2
[5] Título: Neuromancer | Autor: William Gibson | Gênero: Ficção científica | Disponível: 4
[6] Título: Especialista Em Pessoas | Autor: Thiago Brunet | Gênero: Formcao humana baseado em ensinamentos biblios | Disponível: 200

Digite parte do título do livro para buscar: Hobbit

--- Resultados da Busca por 'Hobbit' ---
-> Título: O Hobbit | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 7

Gráfico de Quantidade de Livros por Gênero gerado com sucesso!
(mi_venv) PS D:\Facul>
```