

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).

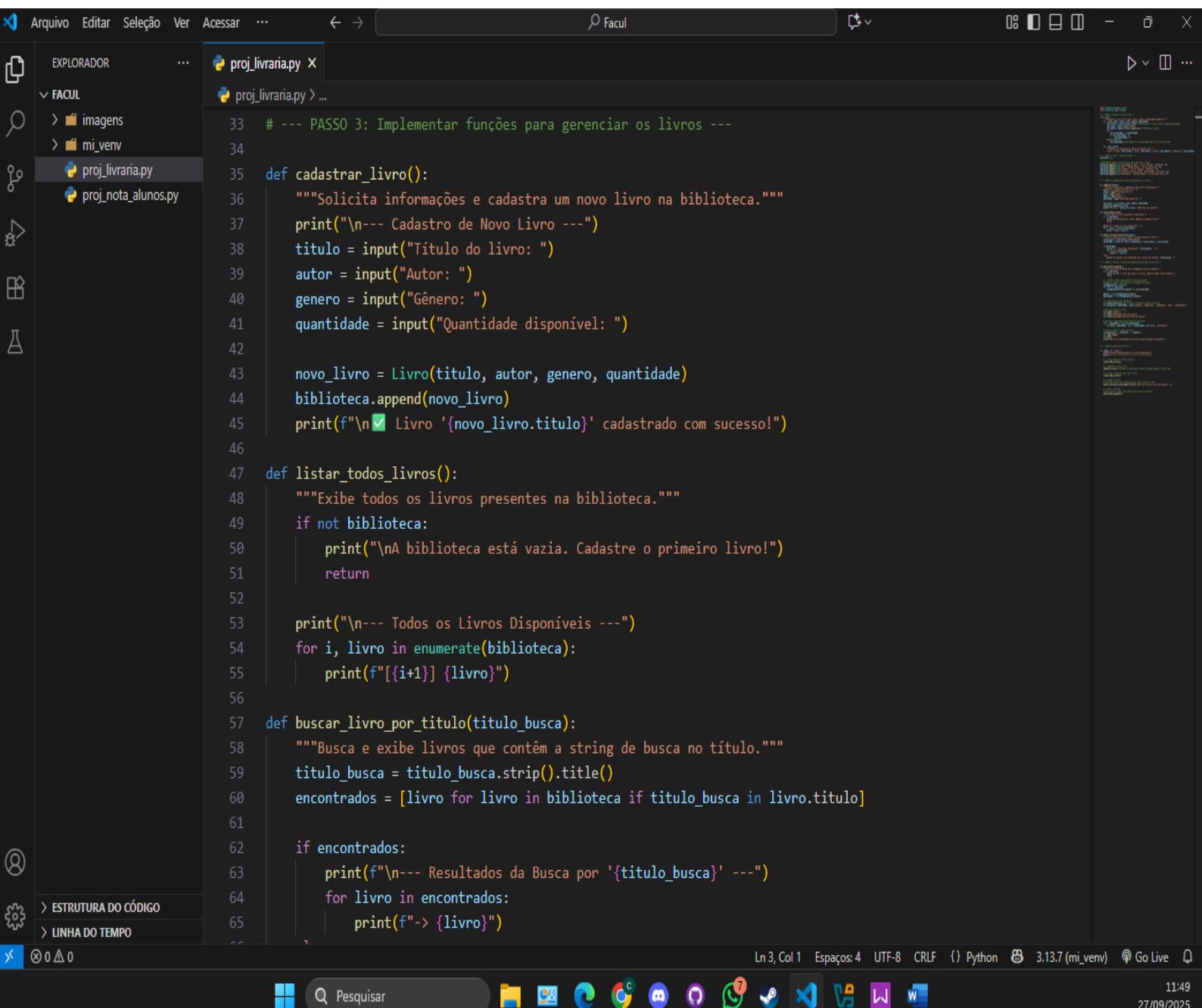
The screenshot shows a Python code editor interface with the following details:

- File Explorer:** Shows a folder structure under "FACUL" containing "imagens", "mi_venv", "proj_livraria.py", and "proj_nota_alunos.py".
- Search Bar:** Contains the text "Facul".
- Code Editor:** Displays the content of "proj_livraria.py". The code defines a "Livro" class with methods for initialization, string representation, and a list of books.
- Output/Console:** On the right side, there is a terminal window showing the results of running the script.
- Status Bar:** Shows file statistics: Line 3, Column 1, Spaces: 4, CRLF, Python 3.13.7 (mi_venv), and a "Go Live" button.

```
proj_livraria.py
1 import matplotlib.pyplot as plt
2 from collections import Counter
3
4 # --- PASSO 1: Definir a classe Livro ---
5
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 ---

Ln 3, Col 1 Espaços: 4 UTF-8 CRLF () Python 3.13.7 (mi_venv) Go Live 11:48 27/09/2025
```

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



The screenshot shows a Python code editor interface with the following details:

- File Explorer:** Shows a folder structure under 'FACUL' containing 'imagens', 'mi_venv', 'proj_livraria.py', and 'proj_nota_alunos.py'. 'proj_livraria.py' is the active file.
- Code Editor:** Displays the content of 'proj_livraria.py'. The code implements CRUD operations for a library (biblioteca).

```
33 # --- PASSO 3: Implementar funções para gerenciar os livros ---
34
35 def cadastrar_livro():
36     """Solicita informações e cadastrá 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     novo_livro = Livro(titulo, autor, genero, quantidade)
43     biblioteca.append(novo_livro)
44     print(f"\n✓ Livro '{novo_livro.titulo}' cadastrado com sucesso!")

45
46 def listar.todos_livros():
47     """Exibe todos os livros presentes na biblioteca."""
48     if not biblioteca:
49         print("\nA biblioteca está vazia. Cadastre o primeiro livro!")
50         return
51
52     print("\n--- Todos os Livros Disponíveis ---")
53     for i, livro in enumerate(biblioteca):
54         print(f"[{i+1}] {livro}")

55
56 def buscar_livro_por_titulo(titulo_busca):
57     """Busca e exibe livros que contêm a string de busca no título."""
58     titulo_busca = titulo_busca.strip().title()
59     encontrados = [livro for livro in biblioteca if titulo_busca in livro.titulo]

60     if encontrados:
61         print(f"\n--- Resultados da Busca por '{titulo_busca}' ---")
62         for livro in encontrados:
63             print(f"--> {livro}")
```
- Status Bar:** Shows the current line (Ln 3, Col 1), character count (Espaços: 4), encoding (UTF-8), and file type (Python).
- Bottom Icons:** Includes icons for search, refresh, file operations, and various developer tools.

Módulo de Visualização de Dados

A função `gerar_grafico_genros()` 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 summarizar os dados.

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer (Left):** Shows a project structure under "FACUL" with files: "imagens", "mi_venv", "proj_livraria.py" (selected), and "proj_nota_alunos.py".
- Code Editor (Center):** Displays the Python script "proj_livraria.py". The code defines a function `gerar_grafico_genros()` which generates a bar chart of book genres. It uses `plt.text` to add genre counts above each bar, `plt.grid` for a dashed grid, `plt.tight_layout`, and `plt.show`. It also prints a success message. The script then demonstrates usage with examples 1 through 5.
- Bottom Status Bar:** Shows file path ("Facul"), line/col (Ln 3, Col 1), encoding (UTF-8), file type (Python), version (3.13.7 (mi_venv)), and date (27/09/2025). It also includes icons for search, replace, and other VS Code features.

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.

The screenshot shows a Python development environment with the following details:

- File Explorer:** Shows files like `proj_livraria.py`, `proj_livraria.py > ...`, `proj_livraria.py > ...`, `proj_livraria.py`, and `proj_nota_alunos.py`.
- Code Editor:** Displays the `proj_livraria.py` file content, which defines a `Livro` class with methods for initializing a book with title, author, and genre, and for setting its quantity.
- Terminal:** Shows the output of running the script, including the creation of a new book ("Especialista em Pessoas" by Thiago Brunet) and a list of all available books.
- Figure Window:** A bar chart titled "Quantidade Total de Livros por Gênero" (Total Number of Books by Genre). The chart has three bars: "Fantasia" (12), "Ficção científica" (7), and "Clássico" (200).
- Status Bar:** Shows the current line (Ln 2, Col 32), character count (33 selected), encoding (UTF-8), file type (Python), version (3.13.7 (mi_venv)), and date (27/09/2025).

Instalação do METPLOTLIB:

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. On the left is the Explorer sidebar with a tree view of files and folders. The main area displays a Python script named `proj_livraria.py`. The terminal at the bottom shows a series of failed attempts to install the `metplotlib` package via `pip`, due to a misspelling in the command. The terminal output also shows the successful download and installation of other packages like `matplotlib`, `contourpy`, and `cycler`.

```
proj_livraria.py
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 DEPURAÇÃO TERMINAL PORTAS
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
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_venv) PS D:\Facul> pip install Mateploitlib
ERROR: Could not find a version that satisfies the requirement Mateploitlib (from versions: none)
ERROR: No matching distribution found for Mateploitlib
● (mi_venv) PS D:\Facul> pip install meteploitlib
ERROR: Could not find a version that satisfies the requirement meteploitlib (from versions: none)
ERROR: No matching distribution found for meteploitlib
● (mi_venv) 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_venv) 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_venv) 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)

Ln 127, Col 28 Espaços: 4 UTF-8 CRLF {} Python 3.13.7 (mi_venv) Go Live
11:30
27/09/2023
```

Primeira Função com o Menu aparente:

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a project structure with files: `proj_livraria.py`, `proj_livraria.py ...`, `imagens`, `mi_venv`, and `proj_nota_alunos.py`.
- Code Editor (Center):** Displays the content of `proj_livraria.py`. The code defines a `Livro` class with an `__init__` method that initializes attributes like `titulo`, `autor`, `genero`, and `quantidade`, and handles invalid input for `quantidade`.
- Terminal (Bottom):** Shows the command line output of running the script with Python. It includes the path `D:\Facul\mi_venv\Scripts\python.exe d:/Facul/proj_livraria.py` and the program's functionality for listing books and creating new ones.
- Status Bar (Bottom):** Provides information about the file (`Ln 127, Col 28`), encoding (`UTF-8`), and environment (`Python 3.13.7 (mi_venv)`).
- Activity Bar (Bottom):** Includes icons for search, replace, find, and other development tools.

Cadastro de um novo Livro:

The screenshot shows a Python code editor interface with the following details:

- File Explorer:** Shows files in the project directory:
 - EXPLORADOR
 - FACUL
 - imagens
 - mi_venv
 - proj_livraria.py (selected)
 - proj_nota_alunos.py
- Code Editor:** The file `proj_livraria.py` contains the following code:

```
import matplotlib.pyplot as plt
from collections import Counter

# --- PASSO 1: Definir a classe Livro ---
class Livro:
    """Representa um livro com título, autor, gênero e quantidade disponível."""
    def __init__(self, titulo, autor, genero, quantidade):
        self.titulo = titulo.strip().title() # Garante que o título e autor fiquem padronizados
        self.autor = autor.strip().title()
        self.genero = genero.strip().capitalize() # Padroniza o gênero
        try:
            self.quantidade = int(quantidade)
            if self.quantidade < 0:
                self.quantidade = 0
        except ValueError:
            self.quantidade = 0

    def __str__(self):
        return f'{self.titulo} - {self.autor} - {self.genero} - {self.quantidade}'

    def descrever(self):
        print(f'Livro: {self.titulo} | Autor: {self.autor} | Gênero: {self.genero} | Disponível: {self.quantidade}')


--- Cadastro de Novo Livro ---
Título do livro: Especialista em Pessoas
Autor: Thiago Brunet
Gênero: Formação humana baseado em ensinamentos bíblicos
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: Formação humana baseado em ensinamentos bíblicos | Disponível: 200
```
- Terminal:** Shows the output of the script execution.
- Bottom Bar:** Includes icons for file operations, search, and various tools.

Função -> Buscar por partes do Titulo do Livro:

The screenshot shows a Python code editor interface with the following details:

- File Explorer:** Shows a folder named "FACUL" containing files: "imagens", "mi_venv", "proj_livraria.py" (selected), and "proj_nota_alunos.py".
- Code Editor:** The file "proj_livraria.py" is open, displaying the following code:

```
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
16    PROBLEMAS   SAÍDA   CONSOLE DE DEPURAÇÃO   TERMINAL   PORTAS
17
18 [1] Título: A Sociedade Do Anel | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 5
19 [2] Título: 1984 | Autor: George Orwell | Gênero: Ficção científica | Disponível: 3
20 [3] Título: O Hobbit | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 7
21 [4] Título: O Retrato De Dorian Gray | Autor: Oscar Wilde | Gênero: Clássico | Disponível: 2
22 [5] Título: Neuromancer | Autor: William Gibson | Gênero: Ficção científica | Disponível: 4
23
24 --- Cadastro de Novo Livro ---
25 Título do livro: Especialista em Pessoas
26 Autor: Thiago Brunet
27 Gênero: Formação Humana baseado em ensinamentos bíblicos
28 Quantidade disponível: 200
29
30     Livro 'Especialista Em Pessoas' cadastrado com sucesso!
31
32 --- Todos os Livros Disponíveis ---
33 [1] Título: A Sociedade Do Anel | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 5
34 [2] Título: 1984 | Autor: George Orwell | Gênero: Ficção científica | Disponível: 3
35 [3] Título: O Hobbit | Autor: J.R.R. Tolkien | Gênero: Fantasia | Disponível: 7
36 [4] Título: O Retrato De Dorian Gray | Autor: Oscar Wilde | Gênero: Clássico | Disponível: 2
37 [5] Título: Neuromancer | Autor: William Gibson | Gênero: Ficção científica | Disponível: 4
38 [6] Título: Especialista Em Pessoas | Autor: Thiago Brunet | Gênero: Formação humana baseado em ensinamentos bíblicos | Disponível: 200
39
40 Digite parte do título do livro para buscar: 
```

- Terminal:** Shows the output of running the script, including the creation of a new book entry and a list of all available books.
- Status Bar:** Displays file statistics (Ln 127, Col 28), encoding (UTF-8), line endings (CRLF), Python version (3.13.7), environment (mi_venv), and a timestamp (11:34 27/09/2025).

Mensagem de Sucesso pelo Gráfico Gerado:

The screenshot shows a Python code editor interface with a dark theme. On the left is a sidebar with icons for file explorer, search, and other tools. The main area has a title bar 'Facul' and a status bar at the bottom.

Code Area:

```
proj_livraria.py
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:

```
Sistema de Gerenciamento de Livros da Biblioteca
Título do livro: Especialista em Pessoas
Autor: Thiago Brunet
Gênero: Formção 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: Formção 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!
```

Status Bar:

Ln 2, Col 32 (33 selecionado) Espaços: 4 UTF-8 CRLF () Python 3.13.7 (mi_venv) Go Live 11:37 27/09/2025