

Projeto Livros

Visão Geral

O projeto Livros é uma aplicação Python projetada para gerenciar uma coleção de livros. Ele permite aos usuários cadastrar novos livros, buscar livros existentes por título ou ISBN, atualizar as quantidades em estoque e remover livros da coleção. A aplicação é estruturada usando princípios de orientação a objetos para promover uma melhor organização e manutenibilidade.

Estrutura do Projeto

```
...
Livros
├── __init__.py
├── main.py
├── models
│   ├── __init__.py
│   └── livro.py
├── services
│   ├── __init__.py
│   └── livraria.py
└── README.md
...
```

Componentes

1. `Livros/__init__.py`

Este arquivo marca o diretório como um pacote Python. Pode ser usado para inicializar o pacote ou definir o que é exportado quando o pacote é importado.

2. `Livros/main.py`

Este é o ponto de entrada da aplicação. Ele contém a lógica principal para gerenciar os livros, incluindo funções para cadastrar, buscar, atualizar e remover livros. Essas funções serão refatoradas para uma classe para aplicar os princípios de encapsulamento e design orientado a objetos.

3. `Livros/models/__init__.py`

Este arquivo marca o diretório `models` como um pacote Python. Pode ser usado para inicializar o pacote ou definir o que é exportado quando o pacote é importado.

4. `Livros/models/livro.py`

Este arquivo define a classe `Livro`, que representa um livro com atributos como nome, autor, ISBN, preço e quantidade em estoque. Inclui métodos para exibir informações do livro e gerenciar o estoque.

5. `Livros/services/__init__.py`

Este arquivo marca o diretório `services` como um pacote Python. Pode ser usado para inicializar o pacote ou definir o que é exportado quando o pacote é importado.

6. `Livros/services/livraria.py`

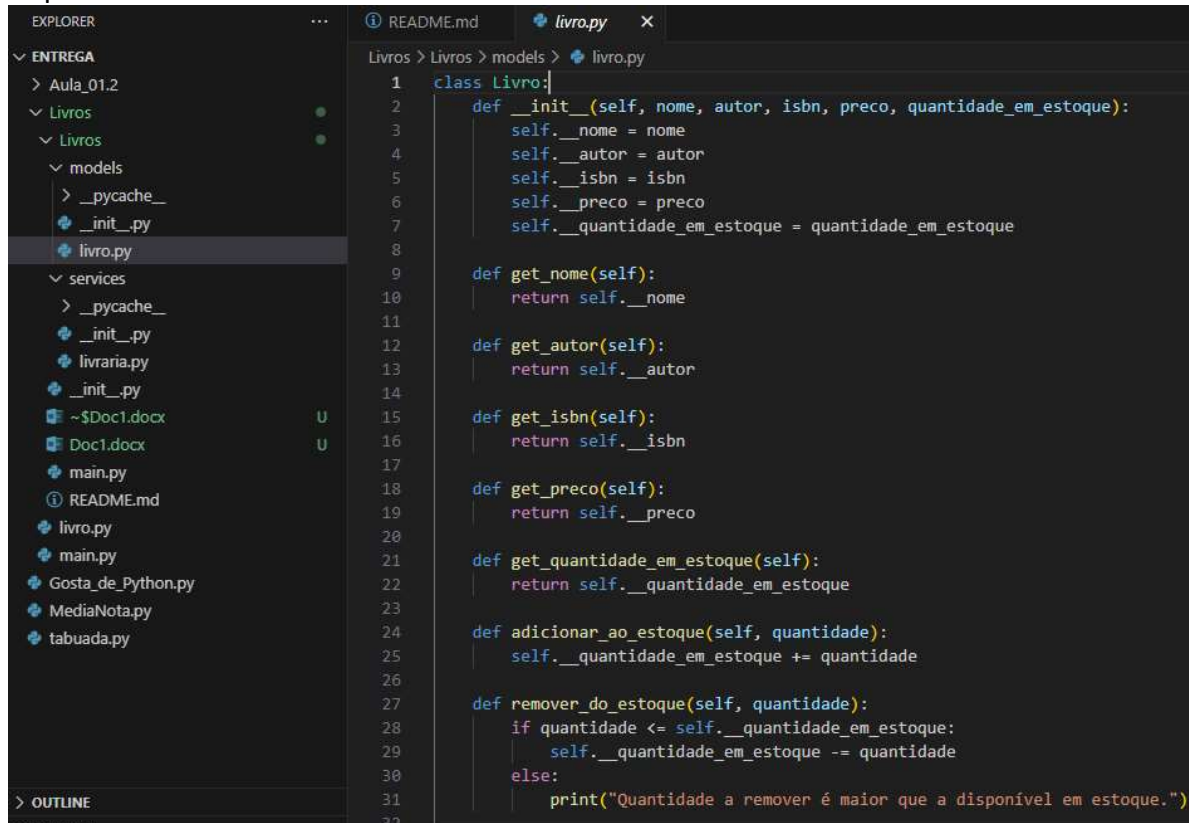
Este arquivo conterá uma nova classe (ex: `Livraria`) que encapsula as funções de gerenciamento de livros do `main.py`. Fornecerá métodos para adicionar, buscar, atualizar e remover livros, promovendo uma melhor organização e encapsulamento.

Uso

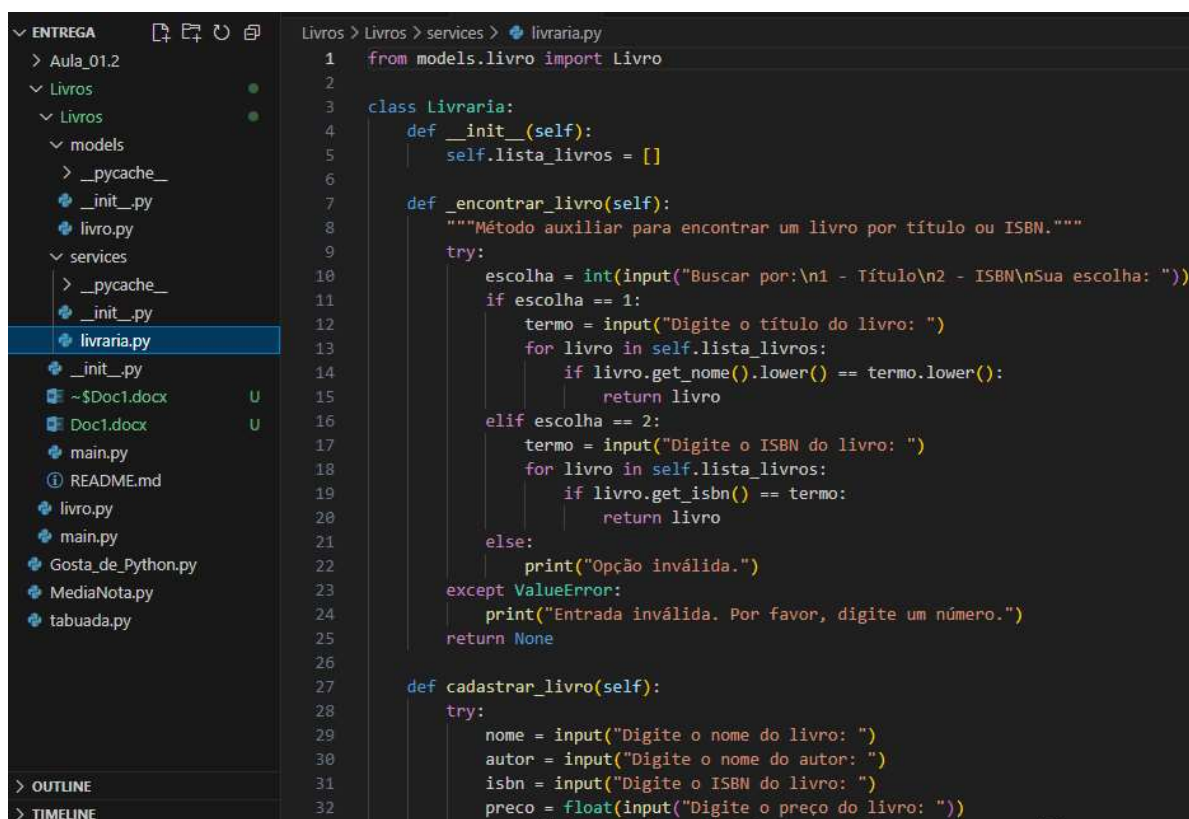
Para executar a aplicação, execute o arquivo `main.py`. Siga o menu na tela para gerenciar a coleção de livros. Certifique-se de que as dependências necessárias estão instaladas e que a estrutura do projeto é mantida.

Melhorias Futuras

- Implementar um banco de dados para armazenamento persistente das informações dos livros.
- Adicionar autenticação de usuário para gerenciar o acesso à coleção de livros.
- Aprimorar a interface do usuário



```
1 class Livro:
2     def __init__(self, nome, autor, isbn, preco, quantidade_em_estoque):
3         self.__nome = nome
4         self.__autor = autor
5         self.__isbn = isbn
6         self.__preco = preco
7         self.__quantidade_em_estoque = quantidade_em_estoque
8
9     def get_nome(self):
10        return self.__nome
11
12    def get_autor(self):
13        return self.__autor
14
15    def get_isbn(self):
16        return self.__isbn
17
18    def get_preco(self):
19        return self.__preco
20
21    def get_quantidade_em_estoque(self):
22        return self.__quantidade_em_estoque
23
24    def adicionar_ao_estoque(self, quantidade):
25        self.__quantidade_em_estoque += quantidade
26
27    def remover_do_estoque(self, quantidade):
28        if quantidade <= self.__quantidade_em_estoque:
29            self.__quantidade_em_estoque -= quantidade
30        else:
31            print("Quantidade a remover é maior que a disponível em estoque.")
32
```



```
1 from models.livro import Livro
2
3 class Livraria:
4     def __init__(self):
5         self.lista_livros = []
6
7     def encontrar_livro(self):
8         """Método auxiliar para encontrar um livro por título ou ISBN."""
9         try:
10             escolha = int(input("Buscar por:\n1 - Título\n2 - ISBN\nSua escolha: "))
11             if escolha == 1:
12                 termo = input("Digite o título do livro: ")
13                 for livro in self.lista_livros:
14                     if livro.get_nome().lower() == termo.lower():
15                         return livro
16             elif escolha == 2:
17                 termo = input("Digite o ISBN do livro: ")
18                 for livro in self.lista_livros:
19                     if livro.get_isbn() == termo:
20                         return livro
21             else:
22                 print("Opção inválida.")
23         except ValueError:
24             print("Entrada inválida. Por favor, digite um número.")
25         return None
26
27     def cadastrar_livro(self):
28         try:
29             nome = input("Digite o nome do livro: ")
30             autor = input("Digite o nome do autor: ")
31             isbn = input("Digite o ISBN do livro: ")
32             preco = float(input("Digite o preço do livro: "))
```

EXPLORER

ENTREGA

Aula_01.2

Livros

Livros

models

__pycache__

init.py

livro.py

services

__pycache__

init.py

livraria.py

init.py

~\$Doc1.docx

Doc1.docx

main.py

README.md

livro.py

main.py

Gosta_de_Python.py

MediaNota.py

tabuada.py

README.md

main.py

Livros > Livros > main.py

1 from services.livraria import Livraria

2

3 def main():

4 livraria = Livraria()

5 livraria.run()

6

7 if __name__ == "__main__":

8 main()

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

PS C:\Julio\14-8\Python_BackEnd_Senai\Exercicios1\entrega\Livros\Livros> python main.py

=====

MENU LIVRARIA

=====

1 - Cadastrar livro

2 - Buscar livro

3 - Atualizar o estoque

4 - Remover Livro

5 - Sair

=====

Sua escolha: