

Cadastrar Livros

A função `CadastrarLivro()` tem como objetivo **cadastrar um novo livro em um sistema**, armazenando suas informações em um banco de dados. Vamos detalhar cada parte da função para facilitar o entendimento:

1. Limpar a tela e solicitar dados do livro

```
os.system("cls")
```

```
print("Insira os dados do novo livro:")
```

- `os.system("cls")`: Limpa a tela do terminal (comum em sistemas Windows).
 - `print("Insira os dados do novo livro:")`: Exibe uma mensagem solicitando os dados do livro.
-

2. Coleta dos dados do livro

A função solicita os seguintes dados do livro:

- **Título**: `titulo = input("Título do Livro: ")`
- **Gênero**: `genero = input("Gênero: ")`
- **Sinopse**: `sinopse = input("Sinopse: ")`
- **Autor**: `autor = input("Autor: ")`
- **Data de publicação**: `publicacao = input("Data de publicação (AAAA/MM/DD): ")`
- **Valor de compra**: `valorCompra = input("Valor de Compra: ")`
- **Valor de revenda**: `valorRevenda = input("Valor de revenda: ")`

Após cada entrada, a função verifica se o usuário digitou "sair" (em letras minúsculas). Se sim, a função redireciona para o `Menu()`.

3. Validação dos dados

```
if not titulo or not genero or not sinopse or not autor or not publicacao or not valorCompra  
or not valorRevenda or len(publicacao) != 10:
```

```
    print("\033[31mDados inválidos. Verifique os dados e tente novamente.\033[m")
```

```
    input()
```

```
    Menu()
```

- Verifica se todos os campos foram preenchidos.

- Confere se a data de publicação (publicacao) tem exatamente 10 caracteres (formato AAAA/MM/DD).
 - Se algum dado estiver inválido, exibe uma mensagem de erro em vermelho (\033[31m) e retorna ao Menu().
-

4. Inserção dos dados no banco de dados

try:

```
cursor.execute(f"INSERT INTO livro (titulo, genero, sinopse, autor, publicacao, valor_compra, valor_revenda) VALUES ('{titulo}', '{genero}', '{sinopse}', '{autor}', '{publicacao}', '{valorCompra}', '{valorRevenda}')
```

```
conexao.commit()
```

- Usa um cursor (cursor) para executar um comando SQL que insere os dados do livro na tabela livro.
 - conexao.commit(): Confirma a inserção dos dados no banco de dados.
-

5. Adicionar o livro ao estoque

```
cursor.execute("SELECT id FROM livro")
```

```
add_estoque = cursor.fetchall()
```

```
for linha in add_estoque:
```

```
    id = linha[0]
```

```
cursor.execute(f"INSERT INTO estoque (id_livro, quantidade) VALUES ('{id}', '{0}')
```

```
conexao.commit()
```

- Recupera o id do livro recém-cadastrado na tabela livro.
 - Insere uma nova linha na tabela estoque, associando o id do livro a uma quantidade inicial de 0.
-

6. Mensagem de sucesso ou erro

```
print("\033[32mCadastro registrado com sucesso.\033[m")
```

```
except:
```

```
    print("\033[31mNão foi possível registrar o cadastro. Verifique os dados e tente novamente.\033[m")
```

- Se tudo der certo, exibe uma mensagem de sucesso em verde (\033[32m).
- Se ocorrer algum erro (por exemplo, dados inválidos ou falha no banco de dados), exibe uma mensagem de erro em vermelho (\033[31m).

7. Retorno ao menu

`input()`

`Menu()`

- Aguarda o usuário pressionar Enter (`input()`).
- Retorna ao `Menu()` principal do sistema.

Resumo da funcionalidade

A função `CadastrarLivro()`:

1. Coleta os dados de um novo livro.
2. Valida se os dados estão corretos.
3. Insere os dados na tabela livro do banco de dados.
4. Adiciona o livro ao estoque com quantidade inicial 0.
5. Exibe mensagens de sucesso ou erro.
6. Retorna ao menu principal.

Essa função é útil para gerenciar o cadastro de livros em um sistema de biblioteca ou loja de livros.

Listar Livros

Aqui está a explicação detalhada da função `ListarLivros()`, que tem como objetivo **listar todos os livros cadastrados no sistema**. Vamos analisar cada parte do código para facilitar o entendimento:

1. Limpar a tela

```
os.system("cls")
```

- `os.system("cls")`: Limpa a tela do terminal (comum em sistemas Windows).
-

2. Consulta ao banco de dados

```
cursor.execute("SELECT * FROM livro")
```

```
resultado = cursor.fetchall()
```

- `cursor.execute("SELECT * FROM livro")`: Executa um comando SQL para selecionar todos os registros da tabela livro.
 - `cursor.fetchall()`: Recupera todos os resultados da consulta e armazena na variável `resultado`.
-

3. Verificação de registros

```
if len(resultado) == 0:
```

```
    print("\033[31mNão há registros.\033[m")
```

```
    input()
```

```
    Menu()
```

- Verifica se a lista de resultados (`resultado`) está vazia.
 - Se não houver registros, exibe uma mensagem de erro em vermelho (`\033[31m`) e retorna ao `Menu()`.
-

4. Exibição dos livros cadastrados

```
print("="*50)
```

```
print("Livros Cadastrados")
```

```
print("="*50)
```

```
for linha in resultado:
```

```
print(f"ID: {linha[0]}")
```

```
print(f"Título: {linha[1]}")
```

```
print(f"Gênero: {linha[2]}")
```

```
print(f"Autor: {linha[3]}")
```

```
print(f"Publicação: {linha[4]}")
```

```
print(f"Sinopse: {linha[5]}\n")
```

```
print("-"*50)
```

- Exibe um cabeçalho formatado com = para destacar o título "Livros Cadastrados".
 - Itera sobre cada linha (linha) da lista resultado e exibe os seguintes dados de cada livro:
 - **ID:** Identificador único do livro.
 - **Título:** Nome do livro.
 - **Gênero:** Categoria do livro.
 - **Autor:** Nome do autor.
 - **Publicação:** Data de publicação.
 - **Sinopse:** Breve descrição do livro.
 - Após exibir os dados de um livro, insere uma linha de separação (-).
-

5. Mensagem de conclusão

```
print("\033[32mListagem Concluída com sucesso.\033[m")
```

- Exibe uma mensagem de sucesso em verde (\033[32m) indicando que a listagem foi concluída.
-

6. Retorno ao menu

```
input()
```

```
Menu()
```

- Aguarda o usuário pressionar Enter (input()).
 - Retorna ao Menu() principal do sistema.
-

Resumo da funcionalidade

A função ListarLivros():

1. Limpa a tela.

2. Consulta todos os livros cadastrados na tabela livro do banco de dados.
3. Verifica se há registros. Se não houver, exibe uma mensagem de erro e retorna ao menu.
4. Exibe os dados de todos os livros cadastrados em um formato organizado.
5. Mostra uma mensagem de sucesso ao concluir a listagem.
6. Retorna ao menu principal.

Essa função é útil para visualizar todos os livros cadastrados no sistema, permitindo ao usuário conferir as informações de forma clara e organizada.

Listar Estoque

Aqui está a explicação detalhada da função `ListarEstoque()`, que tem como objetivo **listar o estoque de livros cadastrados no sistema**. Vamos analisar cada parte do código para facilitar o entendimento:

1. Limpar a tela

```
os.system("cls")
```

- `os.system("cls")`: Limpa a tela do terminal (comum em sistemas Windows).
-

2. Consulta ao banco de dados

```
cursor.execute("select l.id, l.titulo, l.valor_compra, l.valor_revenda, e.quantidade from  
livro as l join estoque as e on l.id = e.id_livro")
```

```
resultado = cursor.fetchall()
```

- `cursor.execute(...)`: Executa um comando SQL que seleciona dados das tabelas **livro** e **estoque**.
 - **Tabela livro (l)**: Recupera o id, titulo, valor_compra e valor_revenda do livro.
 - **Tabela estoque (e)**: Recupera a quantidade disponível no estoque.
 - A junção (JOIN) é feita pelo id do livro (`l.id = e.id_livro`).
 - `cursor.fetchall()`: Recupera todos os resultados da consulta e armazena na variável `resultado`.
-

3. Verificação de registros

```
if len(resultado) == 0:
```

```
    print("\033[31mNão há livros registrados para listar um estoque.\033[m")
```

```
    input()
```

```
    Menu()
```

- Verifica se a lista de resultados (`resultado`) está vazia.
 - Se não houver registros, exibe uma mensagem de erro em vermelho (`\033[31m`) e retorna ao `Menu()`.
-

4. Exibição do estoque

```

print("="*50)

print("ESTOQUE")

print("="*50)

for linha in resultado:

    print(f"livro ID: {linha[0]}")

    print(f"Título: {linha[1]}")

    print(f"Preço de compra: {linha[2]}R$")

    print(f"Valor de revenda: {linha[3]}R$")

    print(f"Quantidade disponível: {linha[4]}\n")

print("-"*50)

```

- Exibe um cabeçalho formatado com = para destacar o título "ESTOQUE".
- Itera sobre cada linha (linha) da lista resultado e exibe os seguintes dados de cada livro:
 - **livro ID:** Identificador único do livro.
 - **Título:** Nome do livro.
 - **Preço de compra:** Valor de compra do livro.
 - **Valor de revenda:** Valor de revenda do livro.
 - **Quantidade disponível:** Quantidade do livro em estoque.
- Após exibir os dados de um livro, insere uma linha de separação (-).

5. Mensagem de conclusão

```
print("\033[32mListagem concluída com sucesso.\033[m")
```

- Exibe uma mensagem de sucesso em verde (\033[32m) indicando que a listagem foi concluída.

6. Retorno ao menu

```
input()
```

```
Menu()
```

- Aguarda o usuário pressionar Enter (input()).
- Retorna ao Menu() principal do sistema.

Resumo da funcionalidade

A função `ListarEstoque()`:

1. Limpa a tela.
2. Consulta os dados de estoque dos livros, combinando informações das tabelas livro e estoque.
3. Verifica se há registros. Se não houver, exibe uma mensagem de erro e retorna ao menu.
4. Exibe os dados de estoque de todos os livros cadastrados em um formato organizado.
5. Mostra uma mensagem de sucesso ao concluir a listagem.
6. Retorna ao menu principal.

Essa função é útil para visualizar o estoque de livros, permitindo ao usuário conferir a quantidade disponível, os preços de compra e revenda, e outras informações relevantes.