

Remover Livros

Aqui está a explicação detalhada da função `RemoverLivro()`, que tem como objetivo **remover um livro do banco de dados**. 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. Solicitar o ID do livro

```
print("Insira o ID do livro a ser removido:")
```

```
id = input()
```

- Solicita ao usuário o **ID do livro** que será removido.

3. Verificar se o campo está vazio

```
if not id:
```

```
    RemoverLivro()
```

- Se o campo estiver vazio (`if not id`), a função se chama novamente (`RemoverLivro()`).

4. Verificar se o usuário deseja sair

```
if id.lower() == "sair":
```

```
    Menu()
```

- Se o usuário digitar "sair", a função retorna ao `Menu()`.

5. Verificar se o livro existe

```
cursor.execute(f"SELECT id, titulo, autor FROM livro WHERE id = {id}")
```

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

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

```
    print(f"\033[31mERRO. Nenhum registro encontrado com o id {id}\033[m")
```

```
    input()
```

```
return RemoverLivro()
```

- Executa uma consulta SQL para verificar se o livro com o ID fornecido existe na tabela livro.
 - Se não houver resultados (`len(resultado) == 0`), exibe uma mensagem de erro em vermelho (`\033[31m`) e chama novamente a função `RemoverLivro()`.
-

6. Exibir os dados do livro

```
print("\033[32mRegistro encontrado !\033[m")
```

```
print("-"*100)
```

```
print(f'{'ID':5}{ 'Título':30}{ 'Autor'}')
```

```
print("-"*100)
```

```
for linha in resultado:
```

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

```
    titulo = linha[1]
```

```
    autor = linha[2]
```

```
    print(f"{'id':<5}{ 'titulo':20}{ 'autor':21}")
```

```
    print("-"*100)
```

- Exibe uma mensagem de sucesso em verde (`\033[32m`) indicando que o registro foi encontrado.
 - Formata e exibe os dados do livro em uma tabela:
 - **ID:** Identificador único do livro.
 - **Título:** Título do livro.
 - **Autor:** Autor do livro.
 - Insere uma linha de separação (-) após exibir os dados.
-

7. Confirmar a exclusão

```
print("Deseja realmente \033[31mexcluir\033[m esse registro ? S / N:")
```

```
opcao = input()
```

- Pergunta ao usuário se ele deseja realmente excluir o registro, destacando a palavra "excluir" em vermelho (`\033[31m`).
-

8. Excluir o livro

```
if opcao.lower() == "s":
```

```
    try:
```

```
        cursor.execute(f"DELETE FROM livro WHERE id = {id}")
```

```
        conexao.commit()
```

```
        print("\033[32mRegistro deletado com sucesso.\033[m")
```

```
    input()
```

```
    Menu()
```

```
    except:
```

```
        print("\031[32mNão foi possível deletar o cadastro por um erro sinistro.\031[m")
```

```
        input("Tecle Enter para retornar ao menu: ")
```

```
    Menu()
```

- Se o usuário confirmar a exclusão (opcao.lower() == "s"):
 - Tenta executar o comando SQL para excluir o livro da tabela livro.
 - Se a exclusão for bem-sucedida, exibe uma mensagem de sucesso em verde (\033[32m).
 - Se ocorrer um erro, exibe uma mensagem de erro em vermelho (\033[31m) e retorna ao Menu().

9. Retorno ao menu

```
else:
```

```
    Menu()
```

- Se o usuário não confirmar a exclusão (opcao.lower() != "s"), a função retorna ao Menu().

Resumo da funcionalidade

A função `RemoverLivro()`:

1. Limpa a tela.
2. Solicita o ID do livro a ser removido.
3. Verifica se o campo está vazio ou se o usuário deseja sair.
4. Verifica se o livro existe no banco de dados.
5. Exibe os dados do livro em um formato organizado.
6. Pergunta ao usuário se ele deseja realmente excluir o registro.

7. Tenta excluir o livro do banco de dados e exibe mensagens de sucesso ou erro.
8. Retorna ao menu principal.

Essa função é útil para gerenciar o catálogo de livros, permitindo ao usuário remover registros de forma segura e confirmada.

Atualizar Livros

A função `AtualizarLivro()` é responsável por atualizar as informações de um livro em um banco de dados. Vamos detalhar cada parte da função para facilitar o entendimento:

1. Limpeza da Tela e Entrada do ID do Livro

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

```
print("Insira o ID do livro a ser atualizado:")
```

```
id = input()
```

- **`os.system("cls")`**: Limpa a tela do console para uma nova interação.
- **`print("Insira o ID do livro a ser atualizado:")`**: Solicita ao usuário que insira o ID do livro que deseja atualizar.
- **`id = input()`**: Captura o ID inserido pelo usuário.

2. Verificação do ID

```
if not id:
```

```
    AtualizarLivro()
```

```
if id.lower() == "sair":
```

```
    Menu()
```

- **`if not id::`** Se o usuário não inserir nenhum ID, a função chama a si mesma novamente (`AtualizarLivro()`), solicitando o ID novamente.
- **`if id.lower() == "sair":`** Se o usuário digitar "sair", a função redireciona para o menu principal (`Menu()`).

3. Busca do Livro no Banco de Dados

```
cursor.execute(f"SELECT * FROM livro WHERE id = {id}")
```

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

- **`cursor.execute(...)`**: Executa uma consulta SQL para buscar o livro com o ID fornecido.
- **`resultado = cursor.fetchall()`**: Armazena o resultado da consulta.

4. Verificação do Resultado da Busca

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

```
    print(f"\033[mERRO. Nenhum registro encontrado com o id {id}\033[m")
```

```
    input()
```

```
    return AtualizarLivro()
```

- **if len(resultado) == 0::** Se nenhum livro for encontrado com o ID fornecido, exibe uma mensagem de erro e chama a função novamente.

5. Exibição dos Dados do Livro

```
print("="*50)

print("\033[32mRegistro encontrado !\033[m")

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]}")

    print(f"Preço de compra: {linha[6]:.2f}R$")

    print(f"Preço de Venda: {linha[7]:.2f}R$")

print("-"*50)
```

- **for linha in resultado::** Itera sobre os dados do livro encontrado e exibe cada campo (ID, Título, Gênero, Autor, etc.).

6. Entrada dos Novos Dados

```
novoTitulo = input("Novo título: ")

novoGenero = input("Novo Gênero: ")

novoAutor = input("Novo autor: ")

novaPublicacao = input("Nova data de publicação (AAAA/MM/DD): ")

novaSinopse = input("Nova sinopse: ")

try:

    novoValorCompra = float(input("Novo preço de compra: "))

    novoValorVenda = float(input("Novo preço de revenda: "))

except ValueError:

    print(f"\033[31mNenhum número inserido.\033[m")

    input()

return AtualizarLivro()
```

- **novoTitulo = input("Novo título: ")**: Solicita e captura o novo título do livro.

- **novoGenero = input("Novo Gênero: ")**: Solicita e captura o novo gênero do livro.
- **novoAutor = input("Novo autor: ")**: Solicita e captura o novo autor do livro.
- **novaPublicacao = input("Nova data de publicação (AAAA/MM/DD): ")**: Solicita e captura a nova data de publicação.
- **novaSinopse = input("Nova sinopse: ")**: Solicita e captura a nova sinopse do livro.
- **try::** Tenta capturar os novos preços de compra e revenda como números decimais.
- **except ValueError::** Se o usuário não inserir números válidos, exibe uma mensagem de erro e chama a função novamente.

7. Validação dos Dados

```
if not novoTitulo or not novoGenero or not novoAutor or not novaPublicacao or not
novaSinopse or not novoValorCompra or not novoValorVenda or len(novaPublicacao) != 10
or novoValorCompra < 0 or novoValorVenda < 0:
```

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

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

```
input()
```

```
Menu()
```

- **if not ...:** Verifica se todos os campos foram preenchidos corretamente e se os valores são válidos (por exemplo, datas no formato correto e preços positivos).
- **os.system("cls")**: Limpa a tela.
- **print("\033[31mDados inválidos. Verifique os dados e tente novamente.\033[m")**: Exibe uma mensagem de erro se os dados forem inválidos.
- **Menu()**: Redireciona para o menu principal.

8. Confirmação da Atualização

```
print("\033[33mCONFIRMAÇÃO:\033[m Deseja realmente Atualizar o registro atual ? S/N")
```

```
confirmacao = input()
```

```
if confirmacao.lower() == "s":
```

```
try:
```

```
cursor.execute(f"UPDATE livro SET titulo = '{novoTitulo}', genero = '{novoGenero}', autor
= '{novoAutor}', publicacao = '{novaPublicacao}', sinopse = '{novaSinopse}', valor_Compra
= '{novoValorCompra}', valor_revenda = '{novoValorVenda}' WHERE (id = {id})")
```

```
conexao.commit()
```

```
print("\033[32mRegistro Atualizado com sucesso.\033[m")
```

```
except:
```

```
print("\033[31mNão foi possível atualizar o registro devido a um erro sinistro.\033[m")
```

```
else:
```

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

```
print("\033[32mAtualização cancelada.\033[m")
```

- **print("\033[33mCONFIRMAÇÃO:\033[m Deseja realmente Atualizar o registro atual ? S/N"):** Solicita confirmação do usuário para atualizar o livro.
- **if confirmacao.lower() == "s":** Se o usuário confirmar (digitando "S"), a função tenta atualizar o livro no banco de dados.
- **cursor.execute(...):** Executa o comando SQL para atualizar os dados do livro.
- **conexao.commit():** Confirma a transação no banco de dados.
- **print("\033[32mRegistro Atualizado com sucesso.\033[m"):** Exibe uma mensagem de sucesso.
- **except::** Se ocorrer um erro durante a atualização, exibe uma mensagem de erro.
- **else::** Se o usuário não confirmar a atualização, exibe uma mensagem de cancelamento.

9. Retorno ao Menu Principal

```
input()
```

```
Menu()
```

- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de sucesso ou erro antes de continuar).
- **Menu():** Retorna ao menu principal após a atualização ou cancelamento.

Resumo

A função `AtualizarLivro()` permite ao usuário atualizar as informações de um livro no banco de dados. Ela solicita o ID do livro, busca os dados atuais, permite a entrada de novos dados, valida essas entradas e, após confirmação, atualiza o registro no banco de dados. Se algo der errado, a função informa o erro e permite ao usuário tentar novamente ou cancelar a operação.

Cadastrar Cliente

A função `CadastrarCliente()` é responsável por cadastrar um novo cliente em um banco de dados. Vamos detalhar cada parte da função para facilitar o entendimento:

1. Limpeza da Tela e Entrada dos Dados do Cliente

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

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

```
nome = input("Nome: ")
```

- **`os.system("cls")`**: Limpa a tela do console para uma nova interação.
- **`print("Insira os dados do novo cliente:")`**: Solicita ao usuário que insira os dados do novo cliente.
- **`nome = input("Nome: ")`**: Captura o nome do cliente inserido pelo usuário.

2. Verificação de Saída

```
if nome.lower() == "sair":
```

```
    Menu()
```

```
telefone = input("Telefone (11 dígitos): ")
```

```
if telefone.lower() == "sair":
```

```
    Menu()
```

```
email = input("E-mail: ")
```

```
if email.lower() == "sair":
```

```
    Menu()
```

- **`if nome.lower() == "sair":`**: Se o usuário digitar "sair" no campo do nome, a função redireciona para o menu principal (`Menu()`).
- **`telefone = input("Telefone (11 dígitos): ")`**: Captura o número de telefone do cliente inserido pelo usuário.
- **`if telefone.lower() == "sair":`**: Se o usuário digitar "sair" no campo do telefone, a função redireciona para o menu principal (`Menu()`).
- **`email = input("E-mail: ")`**: Captura o e-mail do cliente inserido pelo usuário.
- **`if email.lower() == "sair":`**: Se o usuário digitar "sair" no campo do e-mail, a função redireciona para o menu principal (`Menu()`).

3. Validação dos Dados

```
if not nome or not telefone or not email or len(telefone) != 11:
```

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

```
input()
```

```
return CadastrarCliente()
```

- **if not nome or not telefone or not email or len(telefone) != 11::** Verifica se todos os campos foram preenchidos e se o telefone tem exatamente 11 dígitos.
- **print("\033[31mDados inválidos. Verifique os dados e tente novamente.\033[m"):** Exibe uma mensagem de erro se os dados forem inválidos.
- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de erro antes de continuar).
- **return CadastrarCliente():** Chama a função novamente para permitir que o usuário insira os dados corretamente.

4. Inserção dos Dados no Banco de Dados

```
try:
```

```
cursor.execute(f"INSERT INTO cliente (nome, telefone, email) VALUES ('{nome}',  
'{telefone}', '{email}')
```

```
conexao.commit()
```

```
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")
```

- **try::** Tenta inserir os dados do cliente no banco de dados.
- **cursor.execute(...):** Executa o comando SQL para inserir os dados do cliente na tabela cliente.
- **conexao.commit():** Confirma a transação no banco de dados.
- **print("\033[32mCadastro registrado com sucesso.\033[m"):** Exibe uma mensagem de sucesso se o cadastro for realizado com sucesso.
- **except::** Se ocorrer um erro durante a inserção, exibe uma mensagem de erro.

5. Retorno ao Menu Principal

```
input()
```

```
Menu()
```

- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de sucesso ou erro antes de continuar).
- **Menu():** Retorna ao menu principal após o cadastro ou erro.

Resumo

A função `CadastrarCliente()` permite ao usuário cadastrar um novo cliente no banco de dados. Ela solicita o nome, telefone e e-mail do cliente, valida esses dados e, se

estiverem corretos, insere o novo cliente no banco de dados. Se algo der errado, a função informa o erro e permite ao usuário tentar novamente. A função também permite ao usuário sair para o menu principal a qualquer momento digitando "sair".

Listar Clientes

A função `ListarClientes()` é responsável por exibir todos os clientes cadastrados em um banco de dados. Vamos detalhar cada parte da função para facilitar o entendimento:

1. Limpeza da Tela

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

- `os.system("cls")`: Limpa a tela do console para uma nova interação.

2. Consulta ao Banco de Dados

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

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

- `cursor.execute("SELECT * FROM cliente")`: Executa uma consulta SQL para buscar todos os registros da tabela cliente.
- `resultado = cursor.fetchall()`: Armazena o resultado da consulta em uma variável chamada resultado.

3. Verificação de Registros

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

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

```
    input()
```

```
    Menu()
```

- `if len(resultado) == 0`:: Verifica se a lista de resultados está vazia, ou seja, se não há clientes cadastrados.
- `print("\033[31mNão há registros.\033[m")`: Exibe uma mensagem informando que não há registros.
- `input()`: Aguarda uma entrada do usuário (para que ele possa ver a mensagem antes de continuar).
- `Menu()`: Redireciona para o menu principal.

4. Exibição dos Dados dos Clientes

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

```
print("Clientes cadastrados")
```

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

```
for linha in resultado:
```

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

```
    print(f"Nome: {linha[1]}")
```

```
print(f"Telefone: {linha[2]}")
```

```
print(f"E_mail: {linha[3]}\n")
```

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

- **print("-"*50):** Exibe uma linha de separação para organizar a visualização.
- **print("Clientes cadastrados"):** Exibe um título indicando que os clientes cadastrados serão listados.
- **for linha in resultado::** Itera sobre cada registro (linha) da lista de resultados.
- **print(f"ID: {linha[0]}"):** Exibe o ID do cliente.
- **print(f"Nome: {linha[1]}"):** Exibe o nome do cliente.
- **print(f"Telefone: {linha[2]}"):** Exibe o telefone do cliente.
- **print(f"E_mail: {linha[3]}\n"):** Exibe o e-mail do cliente.
- **print("-"*50):** Exibe uma linha de separação entre os clientes.

5. Mensagem de Conclusão

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

- **print("\033[32mListagem Concluída com sucesso.\033[m"):** Exibe uma mensagem indicando que a listagem foi concluída com sucesso.

6. Retorno ao Menu Principal

```
input()
```

```
Menu()
```

- **input():** Aguarda uma entrada do usuário (para que ele possa ver a listagem e a mensagem de conclusão antes de continuar).
- **Menu():** Retorna ao menu principal após a listagem.

Resumo

A função `ListarClientes()` busca todos os clientes cadastrados no banco de dados e os exibe de forma organizada na tela. Se não houver clientes cadastrados, a função informa isso ao usuário e retorna ao menu principal. Após exibir a listagem, a função informa que a operação foi concluída com sucesso e retorna ao menu principal. Essa função é útil para visualizar rapidamente todos os clientes registrados no sistema.