

# Remover Cliente

A função `RemoverCliente()` permite que um cliente seja removido do banco de dados com base no seu ID. 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. Isso garante que a tela esteja limpa antes de solicitar o ID do cliente.

## 2. Entrada do ID do Cliente

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

```
id = input()
```

- **`print("Insira o ID do cliente a ser removido:")`**: Solicita o ID do cliente que será removido.
- **`id = input()`**: Captura o ID inserido pelo usuário.

## 3. Verificação de Entrada Vazia ou Saída

```
if not id:
```

```
    RemoverCliente()
```

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

```
    Menu()
```

- **`if not id:`**: Se o ID estiver vazio, a função chama a si mesma novamente para solicitar o ID.
- **`if id.lower() == "sair":`**: Se o usuário digitar "sair", a função redireciona para o menu principal (`Menu()`).

## 4. Busca do Cliente no Banco de Dados

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

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

- **`cursor.execute(f"SELECT * FROM cliente WHERE id = {id}")`**: Executa uma consulta SQL para buscar o cliente com o ID especificado.
- **`resultado = cursor.fetchall()`**: Armazena o resultado da consulta em uma variável chamada `resultado`.

## 5. Verificação de Cliente Encontrado

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

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

```
input()
```

```
return RemoverCliente()
```

- **if len(resultado) == 0::** Verifica se a lista de resultados está vazia, ou seja, se nenhum cliente foi encontrado com o ID especificado.
- **print(f"\033[31mERRO. Nenhum registro encontrado com o id {id}\033[m"):** Exibe uma mensagem de erro informando que nenhum cliente foi encontrado.
- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de erro antes de continuar).
- **return RemoverCliente():** Chama a função novamente para permitir que o usuário insira um ID válido.

## 6. Exibição do Cliente Encontrado

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

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

```
print(f'{'ID':5}{'Nome':20}{'Telefone':21}{'E_mail'}')
```

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

```
for linha in resultado:
```

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

```
    nome = linha[1]
```

```
    telefone = linha[2]
```

```
    email = linha[3]
```

```
    print(f'{'id':<5}{'nome':20}{'telefone':21}{'email'}')
```

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

- **print("\033[32mRegistro encontrado !\033[m"):** Exibe uma mensagem de sucesso indicando que o cliente foi encontrado.
- **print("-"\*100):** Exibe uma linha de separação para melhorar a visualização.
- **print(f'{'ID':5}{'Nome':20}{'Telefone':21}{'E\_mail'}'):** Exibe o cabeçalho da tabela com os campos ID, Nome, Telefone e E-mail.
- **print("-"\*100):** Exibe outra linha de separação.
- **for linha in resultado::** Itera sobre os dados do cliente encontrado.
- **id = linha[0]:** Armazena o ID do cliente.
- **nome = linha[1]:** Armazena o nome do cliente.
- **telefone = linha[2]:** Armazena o telefone do cliente.
- **email = linha[3]:** Armazena o e-mail do cliente.

- **print(f"{id:<5}{nome:20}{telefone:21}{email}"): Exibe os dados do cliente formatados em uma tabela.**
- **print("-"\*100): Exibe uma linha de separação entre os registros.**

## 7. Confirmação de Exclusão

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

```
opcao = input()
```

- **print("Deseja realmente \033[31mexcluir\033[m esse registro ? S / N:"): Solicita confirmação do usuário para excluir o registro.**
- **opcao = input(): Captura a opção escolhida pelo usuário.**

## 8. Exclusão do Cliente

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

```
try:
```

```
    cursor.execute(f"DELETE FROM cliente 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()
```

```
else:
```

```
    Menu()
```

- **if opcao.lower() == "s":** Se o usuário confirmar a exclusão (digitando "S" ou "s"), a função tenta deletar o cliente.
- **try::** Tenta executar o comando SQL para deletar o cliente.
- **cursor.execute(f"DELETE FROM cliente WHERE id = {id}"): Executa o comando SQL para deletar o cliente com o ID especificado.**
- **conexao.commit():** Confirma a transação no banco de dados.
- **print("\033[32mRegistro deletado com sucesso.\033[m): Exibe uma mensagem de sucesso indicando que o cliente foi deletado.**
- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de sucesso antes de continuar).

- **Menu():** Retorna ao menu principal após a exclusão.
- **except::** Se ocorrer um erro durante a exclusão, exibe uma mensagem de erro.
- **print("\031[32mNão foi possível deletar o cadastro por um erro sinistro.\031[m"):** Exibe uma mensagem de erro.
- **input("Tecle Enter para retornar ao menu: "):** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de erro antes de continuar).
- **Menu():** Retorna ao menu principal após o erro.
- **else::** Se o usuário não confirmar a exclusão (digitando qualquer coisa que não seja "S" ou "s"), a função redireciona para o menu principal (Menu()).

## Resumo

A função `RemoverCliente()` permite que um cliente seja removido do banco de dados com base no seu ID. Ela solicita o ID do cliente, verifica se o cliente existe, exibe os detalhes do cliente e solicita confirmação para a exclusão. Se o usuário confirmar, o cliente é deletado do banco de dados. Se algo der errado, a função informa o erro e permite ao usuário tentar novamente. Essa função é útil para manter o banco de dados atualizado, removendo clientes que não são mais necessários.

# Deletar Avaliação

A função `deletarAvaliacao()` permite que uma avaliação seja removida do banco de dados com base no seu ID. 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. Isso garante que a tela esteja limpa antes de solicitar o ID da avaliação.

## 2. Entrada do ID da Avaliação

```
id = input("Insira o ID da avaliação a ser deletada: ")
```

- **`input("Insira o ID da avaliação a ser deletada: ")`**: Solicita o ID da avaliação que será deletada.
- **`id = input()`**: Captura o ID inserido pelo usuário.

## 3. Verificação de Entrada Vazia ou Saída

```
if not id:
```

```
    deletarAvaliacao()
```

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

```
    Menu()
```

- **`if not id:`**: Se o ID estiver vazio, a função chama a si mesma novamente para solicitar o ID.
- **`if id.lower() == "sair":`**: Se o usuário digitar "sair", a função redireciona para o menu principal (`Menu()`).

## 4. Busca da Avaliação no Banco de Dados

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

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

- **`cursor.execute(f"SELECT * FROM avaliacao WHERE id = {id}")`**: Executa uma consulta SQL para buscar a avaliação com o ID especificado.
- **`resultado = cursor.fetchall()`**: Armazena o resultado da consulta em uma variável chamada `resultado`.

## 5. Verificação de Avaliação Encontrada

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

```
    print(f"\033[31mNão há avaliação registrada com o ID {id}\033[m")
```

```
    input()
```

```
deletarAvaliacao()
```

- **if len(resultado) == 0::** Verifica se a lista de resultados está vazia, ou seja, se nenhuma avaliação foi encontrada com o ID especificado.
- **print(f"\033[31mNão há avaliação registrada com o ID {id}\033[m"):** Exibe uma mensagem de erro informando que nenhuma avaliação foi encontrada.
- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de erro antes de continuar).
- **deletarAvaliacao():** Chama a função novamente para permitir que o usuário insira um ID válido.

## 6. Exibição da Avaliação Encontrada

```
for linha in resultado:
```

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

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

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

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

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

```
    print(f"Comentário: {linha[4]}\n")
```

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

- **for linha in resultado::** Itera sobre os dados da avaliação encontrada.
- **print("-"\*50):** Exibe uma linha de separação para melhorar a visualização.
- **print(f"ID: {linha[0]}"):** Exibe o ID da avaliação.
- **print(f"Cliente: {linha[1]}"):** Exibe o nome do cliente que fez a avaliação.
- **print(f"Livro: {linha[2]}"):** Exibe o título do livro que foi avaliado.
- **print(f"Nota: {linha[3]}"):** Exibe a nota dada pelo cliente.
- **print(f"Comentário: {linha[4]}\n"):** Exibe o comentário feito pelo cliente.
- **print("-"\*50):** Exibe uma linha de separação entre as avaliações.

## 7. Confirmação de Exclusão

```
deletar = input("\033[33mCONFIRMAÇÃO\033[mVocê realmente deseja apagar esse  
registro ? (S/N): ")
```

- **input("\033[33mCONFIRMAÇÃO\033[mVocê realmente deseja apagar esse registro ? (S/N): "):** Solicita confirmação do usuário para excluir a avaliação.
- **deletar = input():** Captura a opção escolhida pelo usuário.

## 8. Exclusão da Avaliação

```

if deletar.lower() == "s":
    try:
        cursor.execute(f"DELETE FROM avaliacao WHERE id = {id}")
        conexao.commit()
        print("\033[32mRegistro deletado com sucesso.\033[m")
        input()
    except:
        print("\033[31mNão foi possível deletar o registro.\033[m")
        input()
    Menu()

```

- **if deletar.lower() == "s":** Se o usuário confirmar a exclusão (digitando "S" ou "s"), a função tenta deletar a avaliação.
- **try::** Tenta executar o comando SQL para deletar a avaliação.
- **cursor.execute(f"DELETE FROM avaliacao WHERE id = {id}"):** Executa o comando SQL para deletar a avaliação com o ID especificado.
- **conexao.commit():** Confirma a transação no banco de dados.
- **print("\033[32mRegistro deletado com sucesso.\033[m"):** Exibe uma mensagem de sucesso indicando que a avaliação foi deletada.
- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de sucesso antes de continuar).
- **Menu():** Retorna ao menu principal após a exclusão.
- **except::** Se ocorrer um erro durante a exclusão, exibe uma mensagem de erro.
- **print("\033[31mNão foi possível deletar o registro.\033[m"):** Exibe uma mensagem de erro.
- **input():** Aguarda uma entrada do usuário (para que ele possa ver a mensagem de erro antes de continuar).
- **Menu():** Retorna ao menu principal após o erro.

## Resumo

A função `deletarAvaliacao()` permite que uma avaliação seja removida do banco de dados com base no seu ID. Ela solicita o ID da avaliação, verifica se a avaliação existe, exibe os detalhes da avaliação e solicita confirmação para a exclusão. Se o usuário confirmar, a avaliação é deletada do banco de dados. Se algo der errado, a função informa o erro e permite ao usuário tentar novamente. Essa função é útil para manter o banco de dados atualizado, removendo avaliações que não são mais necessárias.

# Encerrar Conexão

A função `EncerrarConexao()` tem como objetivo encerrar de forma segura uma conexão com um banco de dados, limpar a tela do terminal e finalizar o programa. Abaixo está uma explicação detalhada de cada parte da função:

## 1. Limpando a tela do terminal

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

O que faz: Executa o comando `cls`, que limpa a tela no terminal do Windows.

Motivo: Deixa a interface mais limpa para o usuário antes de finalizar ou exibir mensagens.

## 2. Encerrando a conexão com o banco de dados

```
try:
```

```
    cursor.close()
```

```
    conexao.close()
```

O que faz:

`cursor.close()`: Fecha o cursor do banco de dados. O cursor é usado para executar comandos SQL e recuperar dados.

`conexao.close()`: Encerra a conexão com o banco de dados para liberar os recursos.

Motivo: É uma boa prática fechar as conexões com o banco de dados para evitar vazamento de recursos ou erros futuros.

## 3. Exibindo mensagem de sucesso

```
print("\033[32mConexão Encerrada com segurança.\033[m")
```

O que faz: Imprime uma mensagem informando que a conexão foi encerrada com sucesso.

Detalhes:

`\033[32m`: Código ANSI que muda a cor do texto para verde.

`\033[m`: Reseta as cores do texto após a mensagem.

Motivo: Fornece um feedback visual ao usuário de que o processo foi bem-sucedido.

## 4. Finalizando o programa

```
os._exit(0)
```

O que faz: Encerra imediatamente a execução do programa com o código de saída 0 (indica término bem-sucedido).



Motivo: Garante que o programa seja encerrado após a mensagem de sucesso.

#### 5. Tratando falhas ao encerrar a conexão

```
except:
```

```
    print("\033[31mNão foi possível encerrar a conexão. Tente novamente.\033[m")
```

```
    input()
```

```
    Menu()
```

O que faz:

Se ocorrer um erro ao tentar fechar o cursor ou a conexão, uma mensagem de erro é exibida.

\033[31m: Muda a cor do texto para vermelho para indicar falha.

input(): Pausa o programa para que o usuário veja a mensagem.

Menu(): Retorna o usuário para o menu principal do programa.

Motivo: Fornece um feedback em caso de erro e dá ao usuário a oportunidade de tentar novamente.

#### Resumo do comportamento

Limpa a tela para uma interface mais limpa.

Tenta fechar o cursor e a conexão com o banco de dados:

Se der certo, exibe uma mensagem verde e encerra o programa.

Se falhar, exibe uma mensagem vermelha, pausa para o usuário ler e retorna ao menu principal.