# Atualizar Cliente

A função AtualizarCliente() é responsável por atualizar as informações de um cliente já cadastrado no banco de dados. Vamos detalhar cada parte da função para facilitar o entendimento:

**1. Limpeza da Tela e Entrada do ID do Cliente**

```
os.system("cls")

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

id = input()
```

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

**2. Verificação de Entrada Vazia ou Saída**

```
if not id:

    AtualizarCliente()

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()).

**3. 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.

**4. Verificação de Registro Encontrado**

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

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

    input()
```

```
  AtualizarCliente()
```

- **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 registro foi encontrado.
- **input()**: Aguarda uma entrada do usuário (para que ele possa ver a mensagem de erro antes de continuar).
- **AtualizarCliente()**: Chama a função novamente para permitir que o usuário insira um ID válido.

**5. Exibição dos Dados 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 informando que o registro foi encontrado.
- **print("-"*100)**: Exibe uma linha de separação para organizar a visualização.
- **print(f"{'ID':5}{'Nome':20}{'Telefone':21}{'E_mail'}")**: Exibe os cabeçalhos das colunas (ID, Nome, Telefone, E-mail).
- **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.
- **print("-"*100)**: Exibe uma linha de separação entre os dados.

**6. Entrada dos Novos Dados**

```
newNome = input("Novo nome: ")
newTelefone = input("Novo Telefone (11 dígitos): ")
newEmail = input("Novo E-mail: ")
```

- **newNome = input("Novo nome: ")**: Solicita e captura o novo nome do cliente.
- **newTelefone = input("Novo Telefone (11 dígitos): ")**: Solicita e captura o novo telefone do cliente.
- **newEmail = input("Novo E-mail: ")**: Solicita e captura o novo e-mail do cliente.

**7. Validação dos Dados**

```
if not newNome or not newTelefone or not newEmail or len(newTelefone) != 11:
    print("\033[31mDados inválidos. O registro não foi atualizado.\033[m")
    input()
    Menu()
```

- **if not newNome or not newTelefone or not newEmail or len(newTelefone) != 11:**: Verifica se todos os campos foram preenchidos corretamente e se o telefone tem exatamente 11 dígitos.
- **print("\033[31mDados inválidos. O registro não foi atualizado.\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).
- **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 cliente SET nome = '{newNome}', telefone = '{newTelefone}', email = '{newEmail}' 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 cliente.
- **if confirmacao.lower() == "s":**: Se o usuário confirmar (digitando "S"), a função tenta atualizar o cliente no banco de dados.
- **cursor.execute(...)**: Executa o comando SQL para atualizar os dados do cliente.
- **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 AtualizarCliente() permite ao usuário atualizar as informações de um cliente já cadastrado no banco de dados. Ela solicita o ID do cliente, 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.

# Adicionar Avaliações

A função AdicionarAvaliacoes() permite que um cliente cadastrado avalie um livro, adicionando uma nota e um comentário. Vamos detalhar cada parte da função para facilitar o entendimento:

**1. Limpeza da Tela e Entrada do ID do Cliente**

```
os.system("cls")

id_cliente = input("ID do cliente que irá avaliar: ")
```

- **os.system("cls")**: Limpa a tela do console para uma nova interação.
- **id_cliente = input("ID do cliente que irá avaliar: ")**: Solicita e captura o ID do cliente que fará a avaliação.

**2. Verificação de Entrada Vazia ou Saída**

```
if not id_cliente:
    AdicionarAvaliacoes()

if id_cliente.lower() == "sair":
    Menu()
```

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

**3. Busca do Cliente no Banco de Dados**

```
cursor.execute(f"select nome from cliente where id = {id_cliente}")

resultado_cliente = cursor.fetchall()
```

- **cursor.execute(f"select nome from cliente where id = {id_cliente}")**: Executa uma consulta SQL para buscar o nome do cliente com o ID especificado.
- **resultado_cliente = cursor.fetchall()**: Armazena o resultado da consulta em uma variável chamada resultado_cliente.

**4. Verificação de Cliente Encontrado**

```
if len(resultado_cliente) == 0:
    print(f"Não há cliente cadastrado com o id {id_cliente}")
    input()
    return AdicionarAvaliacoes()
```

- **if len(resultado_cliente) == 0:**: Verifica se a lista de resultados está vazia, ou seja, se nenhum cliente foi encontrado com o ID especificado.

- **print(f"Não há cliente cadastrado com o id {id_cliente}")**: Exibe uma mensagem informando que nenhum cliente foi encontrado.
- **input()**: Aguarda uma entrada do usuário (para que ele possa ver a mensagem antes de continuar).
- **return AdicionarAvaliacoes()**: Chama a função novamente para permitir que o usuário insira um ID válido.

**5. Captura do Nome do Cliente**

```
for linha in resultado_cliente:
    nome = linha[0]
```

- **for linha in resultado_cliente:**: Itera sobre os dados do cliente encontrado.
- **nome = linha[0]**: Armazena o nome do cliente.

**6. Entrada do ID do Livro**

```
id_livro = input("ID do livro que será avaliado: ")
if id_livro.lower() == "sair":
    Menu()
```

- **id_livro = input("ID do livro que será avaliado: ")**: Solicita e captura o ID do livro que será avaliado.
- **if id_livro.lower() == "sair":**: Se o usuário digitar "sair", a função redireciona para o menu principal (Menu()).

**7. Busca do Livro no Banco de Dados**

```
cursor.execute(f"select titulo from livro where id = {id_livro}")
resultado_livro = cursor.fetchall()
```

- **cursor.execute(f"select titulo from livro where id = {id_livro}")**: Executa uma consulta SQL para buscar o título do livro com o ID especificado.
- **resultado_livro = cursor.fetchall()**: Armazena o resultado da consulta em uma variável chamada resultado_livro.

**8. Verificação de Livro Encontrado**

```
if len(resultado_livro) == 0:
    print(f"Não há livro registrado com o ID {id_livro}")
    input()
    return AdicionarAvaliacoes()
```

- **if len(resultado_livro) == 0:**: Verifica se a lista de resultados está vazia, ou seja, se nenhum livro foi encontrado com o ID especificado.

- **print(f"Não há livro registrado com o ID {id_livro}")**: Exibe uma mensagem informando que nenhum livro foi encontrado.
- **input()**: Aguarda uma entrada do usuário (para que ele possa ver a mensagem antes de continuar).
- **return AdicionarAvaliacoes()**: Chama a função novamente para permitir que o usuário insira um ID válido.

**9. Captura do Título do Livro**

```
for linha in resultado_livro:
    titulo = linha[0]
```

- **for linha in resultado_livro:**: Itera sobre os dados do livro encontrado.
- **titulo = linha[0]**: Armazena o título do livro.

**10. Entrada da Nota**

```
os.system("cls")
print(f"Qual a nota que o cliente {nome} dará ao livro {titulo} de 1 a 10 ?")
try:
    nota = int(input())
except ValueError:
    print("\033[31mA nota precisa ser um número entre 1 e 10.\033[m")
    input()
    return AdicionarAvaliacoes()
if not nota or nota > 10 or nota < 1:
    print("\033[31mA nota precisa ser um número entre 1 e 10.\033[m")
    input()
    return AdicionarAvaliacoes()
```

- **os.system("cls")**: Limpa a tela do console para uma nova interação.
- **print(f"Qual a nota que o cliente {nome} dará ao livro {titulo} de 1 a 10 ?")**: Solicita a nota que o cliente dará ao livro.
- **try:**: Tenta capturar a nota como um número inteiro.
- **except ValueError:**: Se a nota não for um número, exibe uma mensagem de erro.
- **if not nota or nota > 10 or nota < 1:**: Verifica se a nota está dentro do intervalo válido (1 a 10).
- **print("\033[31mA nota precisa ser um número entre 1 e 10.\033[m")**: Exibe uma mensagem de erro se a nota for inválida.

- **input()**: Aguarda uma entrada do usuário (para que ele possa ver a mensagem de erro antes de continuar).
- **return AdicionarAvaliacoes()**: Chama a função novamente para permitir que o usuário insira uma nota válida.

**11. Entrada do Comentário**

```
os.system("cls")
print(f"Insira o comentário da avaliação:")
comentario = input()
```

- **os.system("cls")**: Limpa a tela do console para uma nova interação.
- **print(f"Insira o comentário da avaliação:")**: Solicita o comentário da avaliação.
- **comentario = input()**: Captura o comentário inserido pelo usuário.

**12. Inserção da Avaliação no Banco de Dados**

```
try:
    cursor.execute(f"insert into avaliacao (comentario, nota, id_livro, id_cliente) values ('{comentario}', '{nota}', '{id_livro}', '{id_cliente}')")
    conexao.commit()
    print("\033[32mComentário adicionado com sucesso.\033[m")
except:
    print("\033[31mNão foi possível adicionar o comentário em questão.\033[m")
```

- **try:**: Tenta inserir a avaliação no banco de dados.
- **cursor.execute(...)**: Executa o comando SQL para inserir o comentário, a nota, o ID do livro e o ID do cliente na tabela avaliacao.
- **conexao.commit()**: Confirma a transação no banco de dados.
- **print("\033[32mComentário adicionado com sucesso.\033[m")**: Exibe uma mensagem de sucesso.
- **except:**: Se ocorrer um erro durante a inserção, exibe uma mensagem de erro.

**13. 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 inserção da avaliação.

**Resumo**

A função AdicionarAvaliacoes() permite que um cliente cadastrado avalie um livro, adicionando uma nota e um comentário. Ela solicita o ID do cliente e do livro, verifica se ambos existem no banco de dados, permite a entrada da nota e do comentário, valida essas entradas e, após confirmação, insere a avaliação no banco de dados. Se algo der errado, a função informa o erro e permite ao usuário tentar novamente. Essa função é útil para coletar feedback dos clientes sobre os livros disponíveis.

# Verificar Atualizações

A função VerificarAvaliacoes() é responsável por exibir todas as avaliações registradas no banco de dados, mostrando informações como o ID da avaliação, o nome do cliente, o título do livro, a nota e o comentário. 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 exibir as avaliações.

**2. Consulta no Banco de Dados**

```
cursor.execute("select a.id, c.nome, l.titulo, a.nota, a.comentario from avaliacao as a JOIN livro as l on l.id = a.id_livro JOIN cliente as c on c.id = a.id_cliente")

resultado = cursor.fetchall()
```

- **cursor.execute(...)**: Executa uma consulta SQL que busca informações das avaliações, juntando as tabelas avaliacao, livro e cliente. A consulta retorna o ID da avaliação (a.id), o nome do cliente (c.nome), o título do livro (l.titulo), a nota (a.nota) e o comentário (a.comentario).
- **resultado = cursor.fetchall()**: Armazena o resultado da consulta em uma variável chamada resultado. Esse resultado é uma lista de tuplas, onde cada tupla representa uma avaliação.

**3. Verificação de Avaliações Encontradas**

```
if len(resultado) == 0:
    print("\033[31mNão há registro.\033[m")
    input()
    Menu()
```

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

**4. Exibição das Avaliações**

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

```
print("AVALIAÇÕES")
print("="*50)
for linha in resultado:
    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)
```

- **print("="*50)**: Exibe uma linha de separação para melhorar a visualização.
- **print("AVALIAÇÕES")**: Exibe o título "AVALIAÇÕES".
- **print("="*50)**: Exibe outra linha de separação.
- **for linha in resultado:**: Itera sobre cada avaliação encontrada no banco de dados.
- **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.

**5. Mensagem de Conclusão**

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

- **print("\033[32mConsulta concluída com sucesso.\033[m")**: Exibe uma mensagem de sucesso indicando que a consulta foi realizada com êxito.

**6. Retorno ao Menu Principal**

```
input()
Menu()
```

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

**Resumo**

A função VerificarAvaliacoes() busca e exibe todas as avaliações registradas no banco de dados, mostrando detalhes como o ID da avaliação, o nome do cliente, o título do livro, a

nota e o comentário. Se não houver avaliações, a função informa o usuário e redireciona para o menu principal. Essa função é útil para visualizar o feedback dos clientes sobre os livros de forma organizada e clara.