

DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON

Tema 3: Aula 06 - Python com Banco
de dados (Parte II).



Prof. Ronaldo Candido
ronaldo.candido@estacio.br



Estácio

2024.2

Objetivos da aula

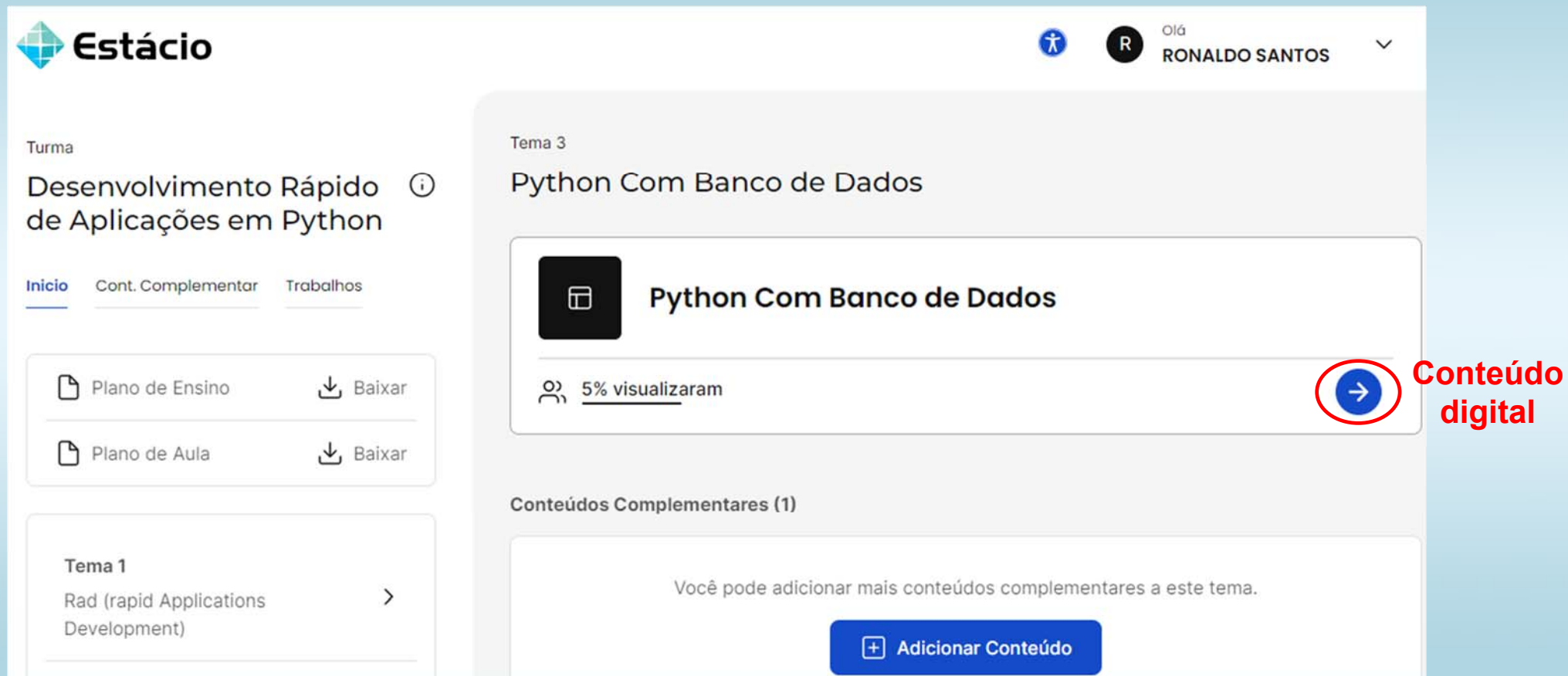
- **Empregar as funcionalidades para conexão, acesso, criação de bancos de dados e tabelas.**

Conteúdo Programático

1. Situação-problema.
2. Conexão ao Banco de dados.
3. Acesso e criação de tabelas.
4. Alterar e excluir tabelas.
5. Atividade verificadora de aprendizagem.
6. Aprenda+.



Sala de Aula Virtual (SAVA)



The screenshot displays the Estácio SAVA interface. At the top, the Estácio logo is on the left, and the user profile 'Olá RONALDO SANTOS' is on the right. The main content area is divided into a left sidebar and a main panel. The sidebar shows the course title 'Desenvolvimento Rápido de Aplicações em Python' with tabs for 'Início', 'Cont. Complementar', and 'Trabalhos'. Below these are download links for 'Plano de Ensino' and 'Plano de Aula'. The main panel shows 'Tema 3: Python Com Banco de Dados' with a document icon, a '5% visualizaram' indicator, and a blue arrow button circled in red. A red label 'Conteúdo digital' points to this button. Below the main panel, there is a section for 'Conteúdos Complementares (1)' with a message and an 'Adicionar Conteúdo' button.

Estácio

Olá
RONALDO SANTOS

Turma

Desenvolvimento Rápido de Aplicações em Python ⓘ

Início Cont. Complementar Trabalhos

Plano de Ensino Baixar

Plano de Aula Baixar

Tema 1

Rad (rapid Applications Development) >

Tema 3

Python Com Banco de Dados

Python Com Banco de Dados

5% visualizaram

→

Conteúdo digital

Conteúdos Complementares (1)

Você pode adicionar mais conteúdos complementares a este tema.

+ Adicionar Conteúdo

Situação-problema

- Nas aulas anteriores, desenvolvemos um aplicativo simples para registro e consulta de notas de alunos em arquivos. Uma desvantagem de usar arquivos é que toda a responsabilidade do gerenciamento dos dados é do programador, que precisará tratar toda a complexidade do ciclo de vida dos dados. Na aula anterior vimos como criar e se conectar a um Banco de Dados usando um SGBD. Para implementar funcionalidades de Bancos de Dados em nosso sistema de registro de notas, precisaremos primeiro criar/manipular as tabelas que armazenarão dados usando o SGBD. **Como realizar esta tarefa usando Python?**



Conexão ao Banco de dados

- Usaremos um script de conexão de banco de dados **SQLite** para criar a estrutura do banco `academia.db` com a tabela cadastro que conterá os campos `codigo`, `nome` e `idade`.
- O Python contém embutido a biblioteca `sqlite3`, bastando importar no início da codificação.
- Mas podemos obter a IDE **SQLiteStudio** <<https://sqlitestudio.pl/>> para manipular diretamente um banco de dados, suas tabelas e relações, além dos dados armazenados. Faça o **download** e extraia o arquivo baixado. Execute o arquivo `sqlitestudio.exe`.



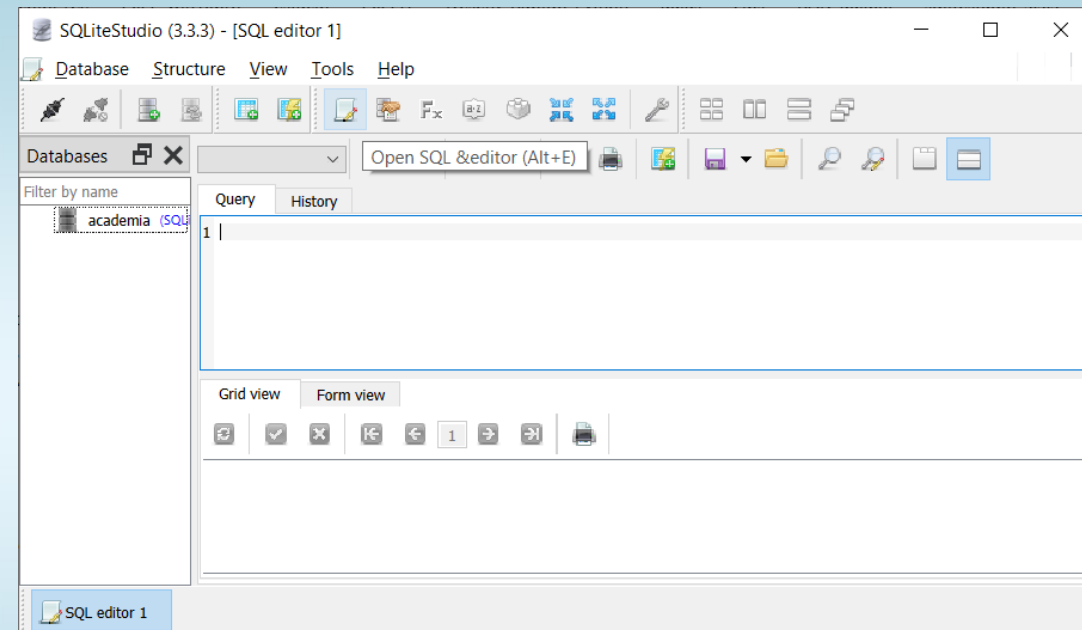
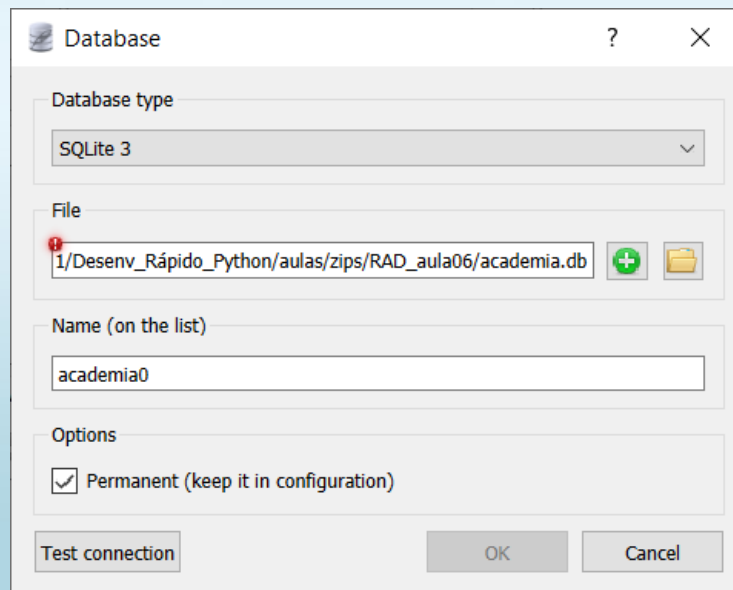
Exemplo de código no SQLite

```
import sqlite3 as conector #apelido
#função para criar a estrutura da tabela
def criar_tabela():
    conexao = conector.connect('academia.db')
    cursor = conexao.cursor()
    #execução de comandos SQL
    sql = "create table cadastro (codigo integer, nome text, idade integer)" #criar tabela
    cursor.execute(sql)
    sql = "insert into cadastro (codigo, nome, idade) values (1284, 'Pedro de Oliveira', 32)" #inserir registros
    cursor.execute(sql)
    sql = "insert into cadastro (codigo, nome, idade) values (1309, 'Lúcia Machado', 37)"
```

```
cursor.execute(sql)
#efetivação do comando
conexao.commit()
#fechamento das conexões
cursor.close()
conexao.close()
#executar função
criar_tabela()
print("Abra a pasta do programa e veja se o arquivo está lá")
print("Fim do programa")
```

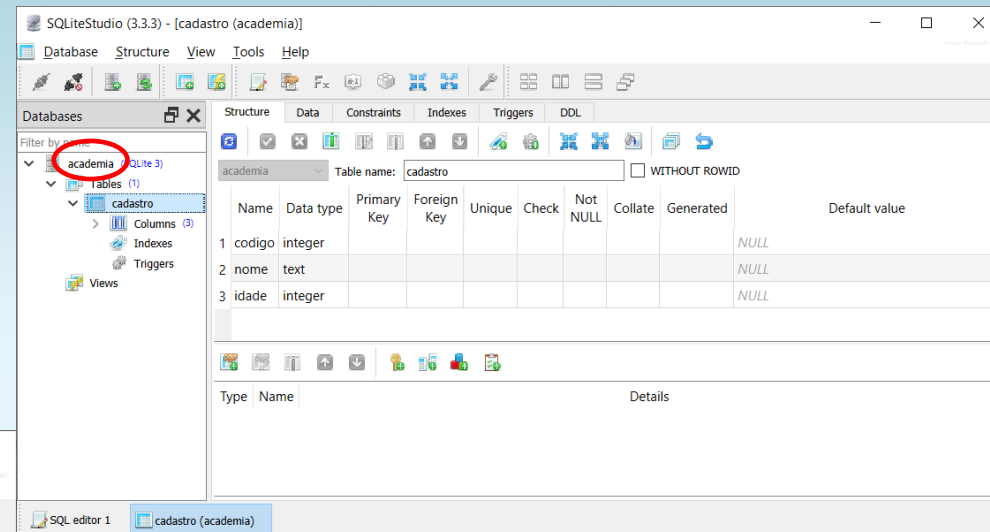
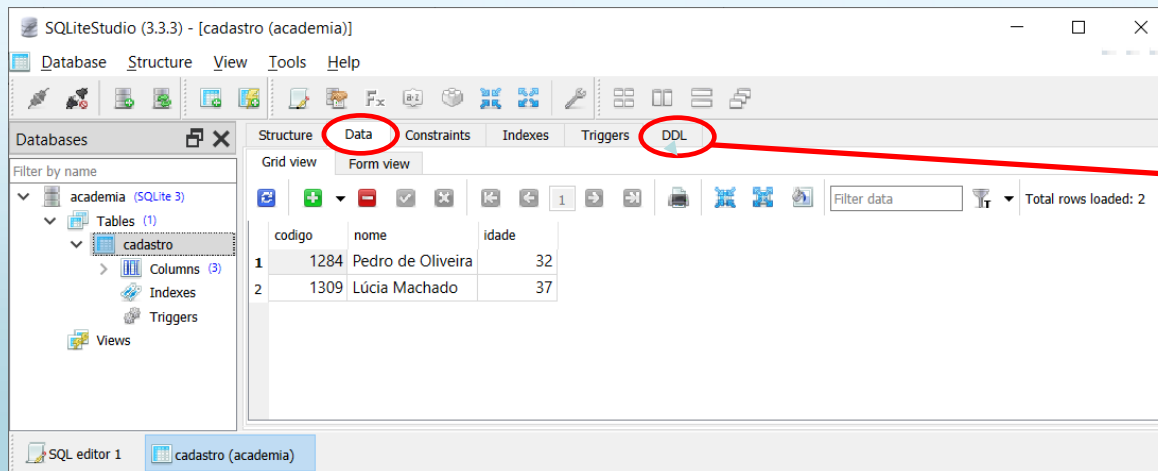
Usando o SQLStudio

- Ao executar o `sqlstudio.exe` será mostrada a seguinte tela ao lado:
- Clicar em **Database / Add a database** para indicar o `academia.db` criado pelo script.



Usando o SQLStudio (continuação)

- Um duplo clique sobre o nome do banco mostrará sua estrutura de tabela e campos.
- Clicar em **Data** mostrará os registros incluídos pelo script.
- Em **DDL** aparece o código **SQL** da estrutura da tabela.



**CREATE TABLE cadastro (
codigo INTEGER,
nome TEXT,
idade INTEGER
);**

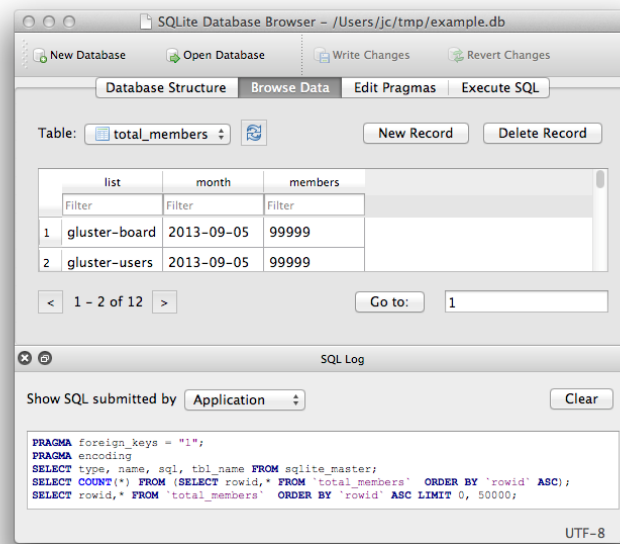
Usando o DB Browser for SQLite

- Outro utilitário de acesso e manutenção de BD para **SQLite** é o **DB Browser**, que pode ser obtido em <https://sqlitebrowser.org/>.

DB Browser for SQLite

The Official home of the DB Browser for SQLite

Screenshot



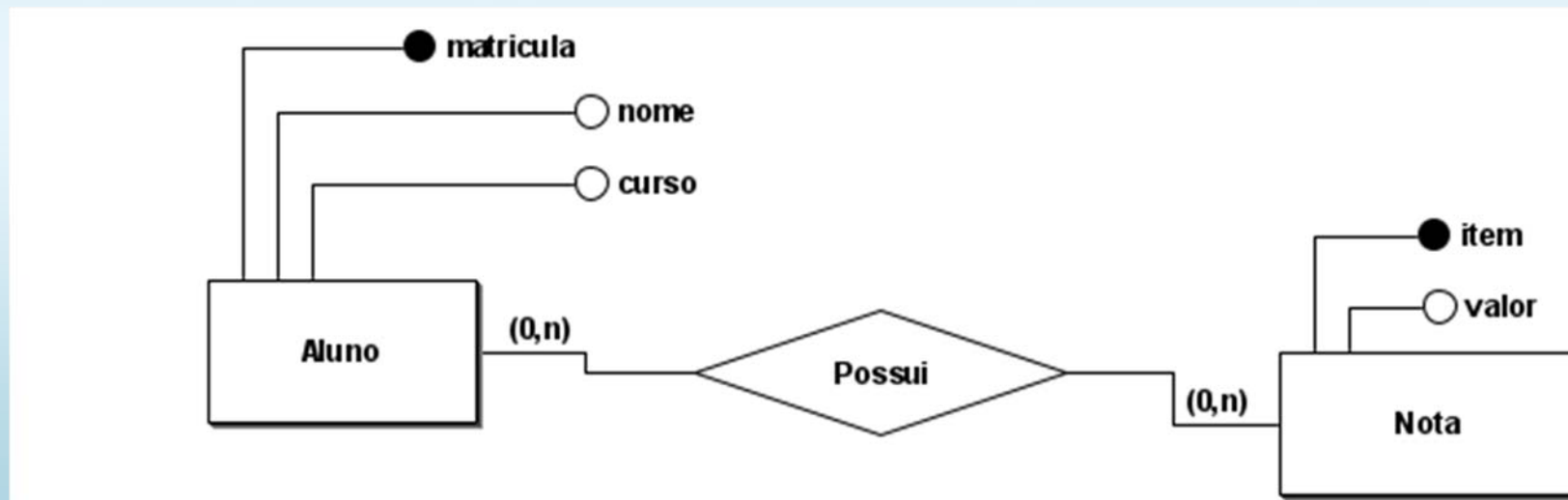
Exemplo de código no SQLite

```
import sqlite3 as conector #apelido
#função para alterar estrutura da tabela
def alterar_tabela():
    print('Abrindo uma conexão de BD')
    conexao = conector.connect('academia.db')
    cursor = conexao.cursor()
    #execução de comandos SQL
    sql = "ALTER TABLE cadastro ADD limite REAL;"
    cursor.execute(sql)
    #atualiza os registros com o campo adicionado
    sql = "UPDATE cadastro SET limite=1000;"
    cursor.execute(sql)
```

```
#efetivação do comando
conexao.commit()
#fechamento das conexões
cursor.close()
conexao.close()
#encerrando
print("Tabela atualizada")
print("Fim do programa")
#executar função
alterar_tabela()
```

Atividade verificadora de aprendizagem

- Voltando ao cenário que trata de um sistema de registro de notas de alunos em uma pequena instituição de ensino, criar as tabelas para modelar as entidades do sistema de registro de notas. Usar o seguinte modelo relacional (BRModelo. Disponível em: <<https://www.sis4.com/brModelo/>>. Acesso em: 18 jul. 2024.



Aprenda +

- Assistir o vídeo: SCHAFER, C. "Python SQLite Tutorial: Complete Overview Creating a Database, Table, and Running Queries". Disponível em <<https://www.youtube.com/watch?v=pd-OGOMigUA>>. (Ative as legendas e a tradução para português). Acesso em: 18 jul. 2024.

- Atividade Autônoma Aura:

Questão 1) O SQLite é uma biblioteca que implementa as funções de gerenciamento de banco de dados de maneira autossuficiente, sem a necessidade de um computador servidor rodando um software servidor de banco de dados. Com o Python instalado, o SQLite também está instalado.

Para abrir uma conexão do Python com o SQLite o código correto é:

- a) `import sqlite3 conector = sqlite3.connect("exemplo.db")`
- b) `import sqlite3 conector = sqlite3.connectar("exemplo.db")`
- c) `import sqlite3 conector = sqlite3.abrir("exemplo.db")`
- d) `import sqlite3 conector = sqlite3.server("exemplo.db")`
- e) `import sqlite3 conector = sqlite3.open("exemplo.db")`

Aprenda + (continuação)

Questão 2) Qual é o comando em SQLite para realizar uma operação de criação de uma tabela no banco de dados ?

- a) commit
- b) close
- c) execute
- d) create
- e) insert



Para a próxima aula...

- Leitura e resolução dos exercícios propostos no livro “**BANIN, S. L. Python 3 Conceitos e Aplicações. Uma Abordagem Didática**”. Capítulo 8: Python 3 com Banco de Dados SQLite.
- Conteúdo digital da disciplina, Tema 3 - “PYTHON COM BANCO DE DADOS”, Módulo 3 “Inserção, remoção e atualização de registros em tabelas”.
- Estudar a Aula 07 de **DESENVOLVIMENTO RÁPIDO EM PYTHON** no SAVA previamente.



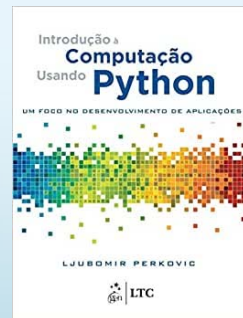
Referências

BANIN, S. L. Python 3 Conceitos e Aplicações. Uma Abordagem Didática. 1a. ed. São Paulo: Érica, 2018. Páginas 312-328. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536530253/>. Acesso em: 16 jul. 2024.

ESTÁCIO. Conteúdo digital da disciplina, Tema "Python com banco de dados", Módulo 2 "Conexão, acesso e criação de bancos de dados e tabelas". Disponível na Sala de aula virtual da disciplina.

PERKOVIC, L. Introdução à Computação Usando Python - Um Foco no Desenvolvimento de Aplicações. 1a. ed. Rio de Janeiro: LTC, 2016. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788521630937/>. Acesso em: 16 jul. 2024.

SALAWA, P. SQLiteStudio. Disponível em: <https://sqlitestudio.pl/>. Acesso em: 18 jul. 2024.



DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON

Dúvidas, sugestões ou análises ???



Prof. Ronaldo Candido
ronaldo.candido@estacio.br



Estácio