



UNIVERSIDADE
FEDERAL DO PIAUÍ

Universidade Federal do Piauí

Disciplina: Engenharia de Software II

Professor: Armando Soares Sousa

Projeto: Microgram

Equipe:

- **Carlos Eduardo Mendes de Oliveira.**
- **José Pires Gayoso**
- **Fernando Vieira Rosa**
- **Markesley Ramos**
- **Bruno Estrella**

Tutorial 3

Tutorial de como instalar e configurar o banco de dados escolhido.

1) Introdução

O Banco de Dados escolhido pela equipe foi o SQLite, que é um sistema de gerenciamento de banco de dados relacional embutido, ou seja, é uma biblioteca que permite a criação e manipulação de bancos de dados relacionais sem a necessidade de um servidor separado. Ele

é muito popular por sua simplicidade, leveza e facilidade de uso, sendo amplamente utilizado em aplicativos de desktop e móveis

Além disso, pelo projeto ser desenvolvido em Python utilizando o framework Flask, utilizaremos o SQLAlchemy, que é uma biblioteca em Python que fornece uma camada de abstração de alto nível para trabalhar com bancos de dados relacionais. Ele suporta vários sistemas de gerenciamento de banco de dados, incluindo SQLite, PostgreSQL, MySQL, entre outros. O SQLAlchemy permite escrever código Python para interagir com bancos de dados de forma mais fácil e flexível, fornecendo recursos avançados, como mapeamento objeto-relacional (ORM) e consultas SQL. Com o SQLAlchemy não precisamos nos preocupar com a criação de códigos SQL, já que tudo é feito inteiramente utilizando Python tornando, assim, a aplicação mais simples.

Mapeamento objeto-relacional ou simplesmente ORM é uma técnica de programação que auxilia na conversão de dados entre banco de dados relacionais e linguagens de programação que são orientadas à objetos.

2) Instalação

- Acesse o site da documentação do Flask-SQLAlchemy [clcando aqui](#) .
- Na Guia do usuário acesse a sessão “**Installation**”.
- No seu terminal ou prompt de comando insira “ `pip install -U Flask-SQLAlchemy` ”

```
$ pip install -U Flask-SQLAlchemy
```

3) Configuração da extensão

- A única configuração necessária do aplicativo Flask é a “SQLALCHEMY_DATABASE_URI” chave. Essa é uma cadeia de conexão que informa ao SQLAlchemy a qual banco de dados se conectar.
- Crie seu objeto de aplicativo Flask, carregue qualquer configuração e inicie a classe de extensão SQLAlchemy com o aplicativo chamando “`db.init_app`”. Este projeto se conecta a um banco de dados SQLite, que é armazenado na pasta da instância do aplicativo.

```
from flask import Flask, render_template, request
from flask_sqlalchemy import SQLAlchemy

#Criando o app
app = Flask(__name__)
#Configurando o SQLite database relativo a pasta da instância do aplicativo
app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///Tutorial03.sqlite3"
#criando a extensão
db = SQLAlchemy()
#Inicializando o app com a extensões
db.init_app(app)
```

- O objeto “db” dá acesso a classe “db.Model” para definir modelos, “db.session” para executar consultas e “db.commit” para confirmar/salvar todas as conexões de banco de dados atuais.

4) Definição dos Modelos

- Para definir uma classe de modelo utiliza-se a Subclasse “db.Model”, que é uma classe base declarativa do SQLAlchemy. Ela receberá atributos “Column” e criará uma tabela.

```
class cursos(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    nome = db.Column(db.String(50))  
    descricao = db.Column(db.String(100))  
    ch = db.Column(db.Integer)
```

- O nome da tabela "cursos" será atribuído automaticamente à tabela do modelo, dessa forma, ao criar a tabela do modelo, ela receberá o nome escolhido para a “class”.

5) Criação de Tabelas

- Depois que todos os modelos e tabelas forem definidos, chame “SQLAlchemy.create_all()” para criar o esquema da tabela no banco de dados. Isso requer um contexto de aplicativo, e como não estamos em uma solicitação neste momento, deve-se criar uma manualmente.
- Ao executar esse comando, será gerado dentro da pasta do projeto um arquivo do tipo “SQLite” que terá o nome colocado na “class”(cursos) no tópico anterior.

```
with app.app_context():  
    db.create_all()
```

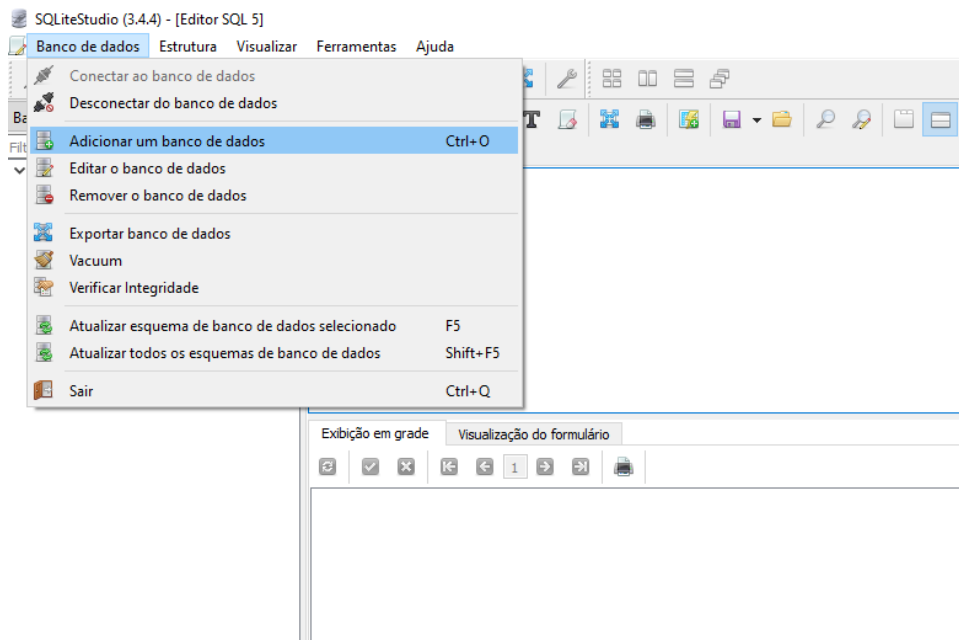
OBS:

- Se você definir modelos em outros módulos, deverá importá-los antes de chamar create_all, caso contrário o SQLAlchemy não saberá sobre eles.
- O “create_all” não atualiza tabelas se elas já estiverem no banco de dados.

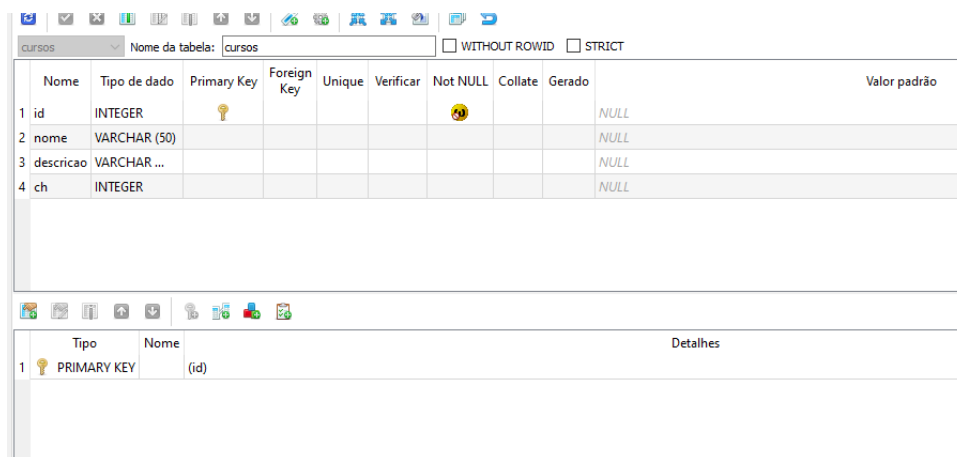
6) Visualização das Tabelas

- Para visualizar a tabela e os campos criados, pode-se acessar o site do [SQLiteStudio](https://sqlitestudio.org/)
- Depois é só apertar em “Download” (para Windows) para baixar.

- Após o download é só executar o instalador baixado e seguir as instruções para finalizar a instalação.
- Abra o SQLiteStudio, clique em “banco de dados” e depois em “adicionar um banco de dados”



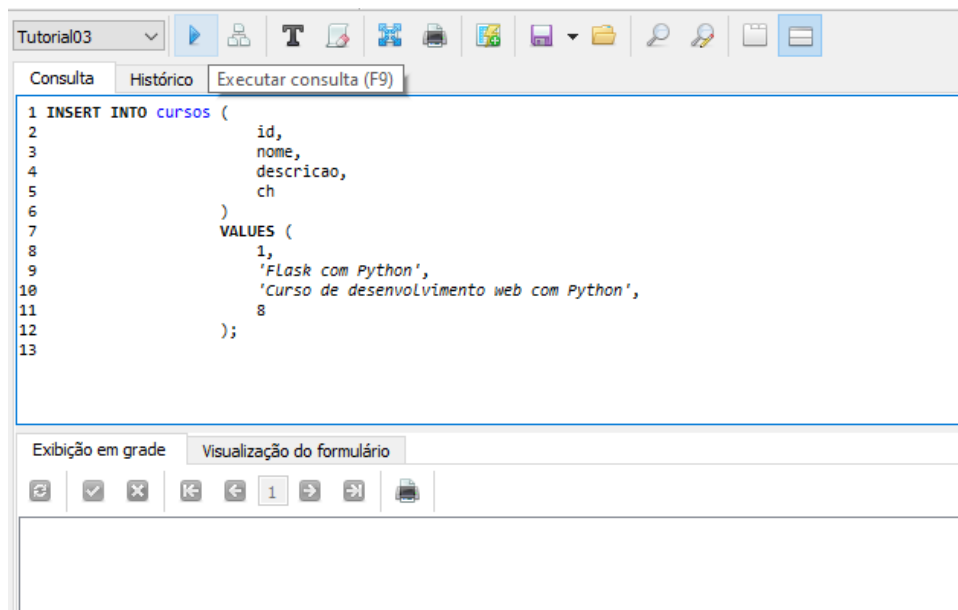
- Depois é preciso preencher no campo “arquivo” o caminho para o arquivo do tipo “SQLite” que foi gerado dentro da pasta do projeto, quando foi executado o comando “create_all()”.
- Após selecionar o arquivo correto, as colunas e campos criados na definição do modelo poderão ser visualizadas dentro do SQLiteStudio



(Foto contendo exemplo)

- Por fim, é possível clicar com botão direito do mouse no nome da sua tabela, e apertar em “gerar consulta para a tabela”, e assim escolher entre: select, insert, delete ou update.

- Inicialmente sua tabela estará vazia então é necessário escolher a função “insert” e preencher com dados nos devidos campos escolhidos para as colunas, e no fim apertar o ícone de “play” para adicionar na tabela.



- É possível preencher os campos quantas vezes quiser, de acordo com o que quiser inserir na tabela, o procedimento sempre será de preencher e apertar o “play” para adicionar.
- Quando quiser visualizar a tabela com as informações já cadastradas, escolha a opção “select” em “adicionar um banco de dados” e pressione o “play”.

The screenshot shows the same database management tool window, but now the 'Visualização do formulário' tab is selected. The toolbar shows 'Total de linhas carregadas: 5'. Below the toolbar, a table is displayed with the following data:

	id	nome	descricao	ch
1	1	Flask com Python	Curso de desenvolvimento web com Python	8
2	2	Django com Python	Curso de desenvolvimento web com Django	10
3	3	Curso de JavaScript	Curso de desenvolvimento web com JavaScript	12
4	4	HTML E CSS	Curso de Frontend	20
5	6	POO com Java	Curso de programação orientada a objeto com Java	12

(Foto contendo exemplo)

Referência para mais detalhes e/ou dúvidas:

- <https://flask-sqlalchemy.palletsprojects.com/en/3.0.x/>

