

# Desenvolvimento de um Site com Flask e Banco de Dados SQLite

Allyson de Almeida Sirvano

20 de maio de 2025

## 1 Introdução

Este documento descreve o desenvolvimento de um site com o microframework Flask, utilizando um banco de dados SQLite. O objetivo deste código é exibir informações sobre livros organizados por categoria, permitindo navegação dinâmica pelas páginas. A aplicação possui uma estrutura simples, usa templates HTML e segue boas práticas de conexão com banco de dados.

## 2 Estrutura do Projeto

A estrutura de diretórios do projeto está organizada da seguinte forma:

```
Site_Flask/  
  
    app.py  
    livros.db  
    templates/  
        index.html  
        livro.html  
        categoria.html  
        500.html  
        404.html
```

## 3 Configuração do Flask e Conexão com o SQLite

O arquivo `app.py` concentra a configuração da aplicação. A seguir, explicamos como a conexão com o banco de dados é feita:

- A aplicação é criada com `Flask(__name__)`;
- A variável `DATABASE` define o caminho para o arquivo do banco de dados;
- A função `get_db()` verifica se já existe uma conexão ativa; se não, cria uma nova com `sqlite3.connect`;
- A opção `row_factory = sqlite3.Row` permite acessar os dados por nome de coluna;

- A função `close_db()` fecha a conexão ao final de cada requisição.

```
from flask import Flask, g, render_template
import sqlite3

app = Flask(__name__)
DATABASE = 'livros.db'

def get_db():
    if 'db' not in g:
        g.db = sqlite3.connect(DATABASE)
        g.db.row_factory = sqlite3.Row
    return g.db

@app.teardown_appcontext
def close_db(error):
    db = g.pop('db', None)
    if db is not None:
        db.close()
```

## 4 Rotas da Aplicação

As rotas determinam como a aplicação responde às URLs. Abaixo estão descritas as principais rotas utilizadas:

### 4.1 Rota Inicial - Página com Lista de Livros

Esta rota responde ao caminho principal (/). Ela recupera todos os livros da base de dados e envia os dados para o template `index.html`:

```
@app.route('/')
def index():
    db = get_db()
    livros = db.execute('SELECT * FROM livros').fetchall()
    return render_template('index.html', livros=livros)
```

### 4.2 Rota Dinâmica - Página de um Livro

A rota `/livro/<int:id>` recebe o ID do livro e mostra seus detalhes usando o template `livro.html`:

```
@app.route('/livro/<int:id>')
def livro(id):
    db = get_db()
    livro = db.execute('SELECT * FROM livros WHERE id = ?', (id,)).
        fetchone()
    return render_template('livro.html', livro=livro)
```

### 4.3 Rota Dinâmica - Página de uma Categoria

Essa rota mostra todos os livros de uma determinada categoria. O nome da categoria é passado na URL:

```
@app.route('/categoria/<nome>')
def categoria(nome):
    db = get_db()
    livros = db.execute('SELECT * FROM livros WHERE categoria = ?', (
        nome,)).fetchall()
    return render_template('categoria.html', categoria=nome, livros=
        livros)
```

## 5 Templates HTML

Template HTML, são essencialmente, uma estrutura base para uma página web, contendo a estrutura básica (HTML) e o estilo visual (CSS) que pode ser personalizado para diferentes conteúdos. É como uma moldura que define a aparência e organização geral de um site ou de uma página, permitindo que se crie conteúdo dentro dessa estrutura pré-definida. Todos os arquivos ficam no diretório `templates/`.

### 5.1 index.html

Lista todos os livros cadastrados. Cada título leva à página do livro e cada categoria à página da categoria correspondente:

```
<!DOCTYPE html>
<html>
<head><title>Livros</title></head>
<body>
    <h1>Lista de Livros</h1>
    <ul>
        {% for livro in livros %}
            <li><a href="/livro/{{ livro.id }}">{{ livro.titulo }}</a> -
                <a href="/categoria/{{ livro.categoria }}">{{ livro.categoria
                    }}</a></li>
        {% endfor %}
    </ul>
</body>
</html>
```

### 5.2 livro.html

Exibe os dados detalhados de um livro:

```
<!DOCTYPE html>
<html>
<head><title>{{ livro.titulo }}</title></head>
<body>
    <h1>{{ livro.titulo }}</h1>
    <p>Autor: {{ livro.autor }}</p>
    <p>Categoria: <a href="/categoria/{{ livro.categoria }}">{{ livro.
        categoria }}</a></p>
    <p>{{ livro.descricao }}</p>
    <a href="/">Voltar</a>
</body>
</html>
```

### 5.3 categoria.html

Mostra todos os livros de uma categoria específica:

```
<!DOCTYPE html>
<html>
<head><title>Categoria: {{ categoria }}</title></head>
<body>
  <h1>Categoria: {{ categoria }}</h1>
  <ul>
    {% for livro in livros %}
      <li><a href="/livro/{{ livro.id }}">{{ livro.titulo }}</a></li>
    {% endfor %}
  </ul>
  <a href="/">Voltar</a>
</body>
</html>
```

## 6 Considerações Finais

Com este código, foi possível desenvolver uma aplicação web simples com Flask e SQLite. Foram abordados os principais pontos como estrutura de diretórios, definição de rotas com parâmetros, conexão com banco de dados e uso de templates para gerar páginas dinâmicas.