**CONFIGURAÇÃO INICIAL**

**1. Instalação**

pip install flask flask-sqlalchemy

**2. Estrutura Básica (app.py)**

from flask import Flask, request, jsonify

from flask\_sqlalchemy import SQLAlchemy

app = Flask(\_\_name\_\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///banco.db'

db = SQLAlchemy(app)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(port=5001, debug=True) # Evita conflito com AirPlay (macOS)

**MODELAGEM DO BANCO DE DADOS**

**1. Tabela Enquete**

class Enquete(db.Model):

id = db.Column(db.Integer, primary\_key=True)

titulo = db.Column(db.String(100), nullable=False)

descricao = db.Column(db.String(200))

data\_criacao = db.Column(db.DateTime, server\_default=db.func.now())

opcoes = db.relationship('Opcao', backref='enquete', lazy=True)

**2. Tabela Opcao**

class Opcao(db.Model):

id = db.Column(db.Integer, primary\_key=True)

texto = db.Column(db.String(100), nullable=False)

votos = db.Column(db.Integer, default=0)

id\_enquete = db.Column(db.Integer, db.ForeignKey('enquete.id'))

**3. Criar Banco**

with app.app\_context():

db.create\_all() # Executar apenas uma vez!

**ENDPOINTS (ROTAS)**

**1. Criar Enquete (POST)**

@app.route('/api/enquetes', methods=['POST'])

def criar\_enquete():

dados = request.get\_json()

nova\_enquete = Enquete(

titulo=dados['titulo'],

descricao=dados.get('descricao', '')

)

db.session.add(nova\_enquete)

db.session.commit()

return jsonify({"mensagem": "Enquete criada!", "id": nova\_enquete.id}), 201

**Teste no Postman**:

* **Método**: POST
* **URL**: http://localhost:5001/api/enquetes
* **Body (JSON)**:

{

"titulo": "Exemplo",

"descricao": "Descrição opcional"

}

**2. Listar Enquetes (GET)**

@app.route('/api/enquetes', methods=['GET'])

def listar\_enquetes():

enquetes = Enquete.query.all()

return jsonify([{"id": e.id, "titulo": e.titulo} for e in enquetes])

**3. Adicionar Opção (POST)**

@app.route('/api/enquetes/<int:id>/opcoes', methods=['POST'])

def adicionar\_opcao(id):

nova\_opcao = Opcao(

texto=request.json['texto'],

id\_enquete=id

)

db.session.add(nova\_opcao)

db.session.commit()

return jsonify({"mensagem": "Opção adicionada!"}), 201

**SOLUÇÃO DE PROBLEMAS COMUNS**

**Erro 403 Forbidden**

* **Causa**: Falta do header Content-Type: application/json.
* **Solução**:

@app.before\_request

def disable\_csrf():

if request.endpoint in ['criar\_enquete', 'adicionar\_opcao']:

request.environ['HTTP\_CONTENT\_TYPE'] = 'application/json'

**Erro "ModuleNotFoundError"**

* **Solução**:

# Verifique se o pacote está instalado

pip show flask-sqlalchemy

# Se não estiver:

pip install flask-sqlalchemy

**ANOTAÇÕES RÁPIDAS**

1. **Portas no macOS**:
   * Evite a porta 5000 (usada pelo AirPlay). Use 5001.
2. **Tipos de Dados SQLAlchemy**:
   * db.Integer, db.String, db.DateTime, db.Boolean
3. **Comandos Úteis**:
   * db.session.add(): Adiciona registro.
   * db.session.commit(): Salva alterações.
   * Model.query.all(): Consulta todos os registros.

**EXTRA: Boas Práticas**

1. **Ambientes Virtuais**:

python -m venv venv

source venv/bin/activate # Linux/Mac

1. **Validação**:  
   Sempre verifique se os campos obrigatórios foram enviados:

if not request.json or 'titulo' not in request.json:

return jsonify({"erro": "Dados inválidos!"}), 400

1. **Documentação**:  
   Use o Postman para gerar documentação automática da API.

**RESUMO FINAL**

* **Modelos**: Definem a estrutura do banco.
* **Endpoints**: Manipulam os dados (CRUD).
* **Postman**: Teste cada rota com método/URL/body corretos.
* **Erros Comuns**: Sempre verifique headers e tipos de dados.