

Configuração do Flask e banco de dados SQLite.

```
app.py
Code > app.py > Produto
1 from flask import Flask, render_template, request, redirect, url_for
2 from flask_sqlalchemy import SQLAlchemy
3 from datetime import datetime
4 import os
5
6 # Configurações Gerais
7 app = Flask(__name__)
8
9 BASE_DIR = os.path.abspath(os.path.dirname(__file__))
10 os.makedirs(os.path.join(BASE_DIR, "database"), exist_ok=True)
11 app.config["SQLALCHEMY_DATABASE_URI"] = (
12     f"sqlite:///os.path.join(BASE_DIR, 'database', 'db-produtos.db')}"
13 )
14
15 app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
16
17 db = SQLAlchemy(app)
18
19 # Entidade do Banco de Dados
20 class Produto(db.Model):
21     __tablename__ = "produtos"
22     idproduto = db.Column(db.Integer, primary_key=True)
23     descricao = db.Column(db.String(100), nullable=False)
24     precocompra = db.Column(db.Float, nullable=False)
25     precovenda = db.Column(db.Float, nullable=False)
26     datacriacao = db.Column(db.Date, nullable=False)
27
28     def __repr__(self):
29         return f"<Produto {self.descricao}>"
30
```

Rotas Flask e suas funções.

```
33 # Rotas.
34 @app.route("/")
35 def home():
36     return redirect(url_for("listar_produtos"))
37
38
39 @app.route("/produtos/cadastrar", methods=["GET", "POST"])
40 def cadastrar_produto():
41     if request.method == "POST":
42         descricao = request.form["descricao"]
43         precocompra = float(request.form["precocompra"])
44         precovenda = float(request.form["precovenda"])
45         datacriacao = datetime.now()
46
47         produto = Produto(
48             descricao=descricao,
49             precocompra=precocompra,
50             precovenda=precovenda,
51             datacriacao=datacriacao,
52         )
53
54         db.session.add(produto)
55         db.session.commit()
56
57         return redirect(url_for("listar_produtos"))
58
59     return render_template("cadastrar.html")
60
61
62 @app.route("/produtos/listar")
63 def listar_produtos():
64     produtos = Produto.query.all()
65     return render_template("listar.html", produtos=produtos)
66
67
68 @app.route("/produtos/editar/<int:idproduto>", methods=["GET", "POST"])
69 def editar_produto(idproduto):
70     produto = Produto.query.get_or_404(idproduto)
71
72     if request.method == "POST":
73         produto.descricao = request.form["descricao"]
74         produto.precocompra = float(request.form["precocompra"])
75         produto.precovenda = float(request.form["precovenda"])
76         db.session.commit()
77         return redirect(url_for("listar_produtos"))
78
79     return render_template("editar.html", produto=produto)
80
81
82 @app.route("/produtos/excluir/<int:idproduto>")
83 def excluir_produto(idproduto):
84     produto = Produto.query.get_or_404(idproduto)
85     db.session.delete(produto)
86     db.session.commit()
87     return redirect(url_for("listar_produtos"))
88
89
90 @app.errorhandler(404)
91 def pagina_nao_encontrada(e):
92     return render_template("404.html"), 404
93
94
95 if __name__ == "__main__":
96     with app.app_context():
97         db.create_all()
98         app.run(debug=True)
```

Modelo Produto definido para o SQLAlchemy.

```
models.py
Code > models.py > ...
1  from Code.app import db
2
3
4  class Produto(db.Model):
5      __tablename__ = "produtos"
6
7      idproduto = db.Column(db.Integer, primary_key=True)
8      descricao = db.Column(db.String(100), nullable=False)
9      precocompra = db.Column(db.Float, nullable=False)
10     precovenda = db.Column(db.Float, nullable=False)
11     datacriacao = db.Column(db.Date, nullable=False)
12
13     def __repr__(self):
14         return f"<Produto {self.descricao}>"
15
```

Formulário HTML para o cadastro de produtos.

```
58 <body>
59 <div class="container">
60     <h2>Cadastrar Produto</h2>
61     <form method="post">
62         <input name="descricao" placeholder="Descrição" required>
63         <input name="precocompra" type="number" step="0.01" placeholder="Preço de Compra" required>
64         <input name="precovenda" type="number" step="0.01" placeholder="Preço de Venda" required>
65         <button type="submit">Salvar</button>
66     </form>
67     <a href="/produtos/listar">Listar Produtos</a>
68 </div>
69 </body>
70
71 </html>
```

Banco de dados SQLite demonstrando a tabela de produtos.



The screenshot shows a SQLite database interface with a dark theme. At the top, there are tabs for 'app.py' and 'db-produtos.db'. Below the tabs, the breadcrumb 'Code > database > db-produtos.db' is visible. The main area displays a query 'SELECT * FROM produtos' with a dropdown menu showing 'produtos'. To the right of the query are buttons for 'Schema', 'Query Editor', 'Auto Reload', 'Find', and 'Other Tools...'. Below the query, the table schema is shown: 'idproduto INTEGER NOT NULL PRIMARY KEY', 'descricao VARCHAR(100) NOT NULL', 'precocompra FLOAT NOT NULL', 'precovenda FLOAT NOT NULL', and 'datacriacao DATE NOT NULL'. At the bottom, a table of data is displayed with one row containing the values 1, Descrição do Produt..., 12.0, 29.0, and 2025-11-05.

	idproduto INTEGER NOT NULL PRIMARY KEY	descricao VARCHAR(100) NOT NULL	precocompra FLOAT NOT NULL	precovenda FLOAT NOT NULL	datacriacao DATE NOT NULL
1	1	Descrição do Produt...	12.0	29.0	2025-11-05
+					