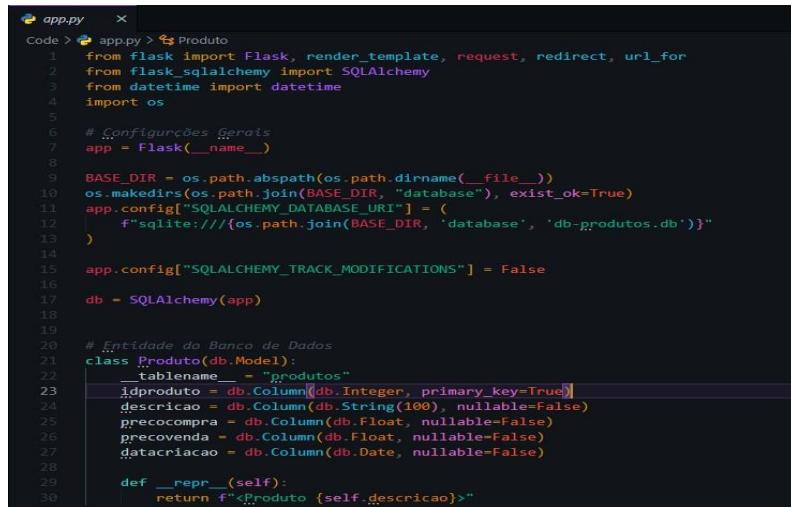
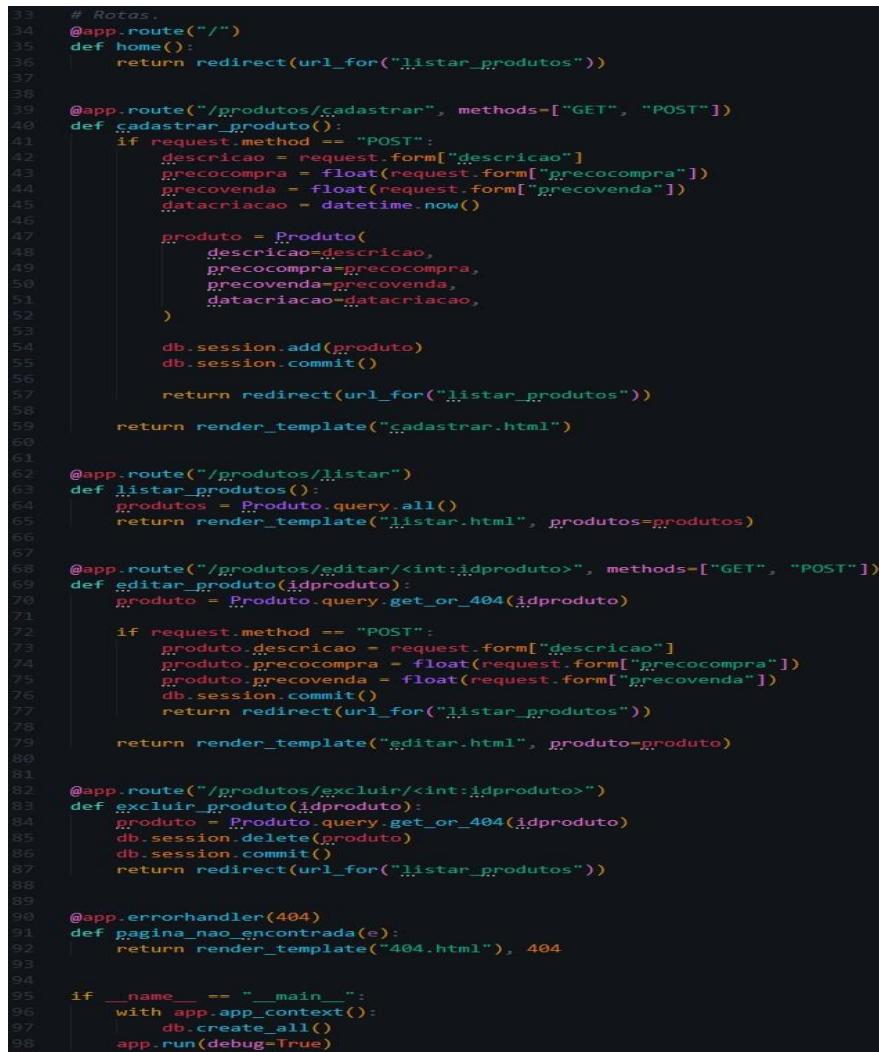


app.py - Configuração principal do Flask e banco de dados SQLite.



```
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
20 # Entidade do Banco de Dados
21 class Produto(db.Model):
22     __tablename__ = "produtos"
23     idproduto = db.Column(db.Integer, primary_key=True)
24     descricao = db.Column(db.String(100), nullable=False)
25     precocompra = db.Column(db.Float, nullable=False)
26     precovenda = db.Column(db.Float, nullable=False)
27     datacriacao = db.Column(db.Date, nullable=False)
28
29     def __repr__(self):
30         return f"Produto {self.descricao}"
31
```

app.py Rotas Flask e Funções.



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

models.py - 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
```

cadastrar.html - Formulário HTML para 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>
```

db-produtos.db - Banco de dados SQLite mostrando tabela de produtos.

The screenshot shows a SQLite database interface with the following details:

- Top bar: app.py, db-produtos.db X
- Navigation: Code > database > db-produtos.db
- Toolbar: SELECT * FROM produtos (highlighted), Schema, Query Editor, Auto Reload, Find, Other Tools...
- Table structure:

	<code>idproduto INTEGER NOT NULL PRIMARY KEY</code>	<code>descricao VARCHAR(100)</code>	<code>precocompra FLOAT NOT NULL</code>	<code>precovenda FLOAT NOT NULL</code>	<code>datacriacao DATE NOT NULL</code>
1	1	Descrição do Produto	12.0	29.0	2025-11-05
+					