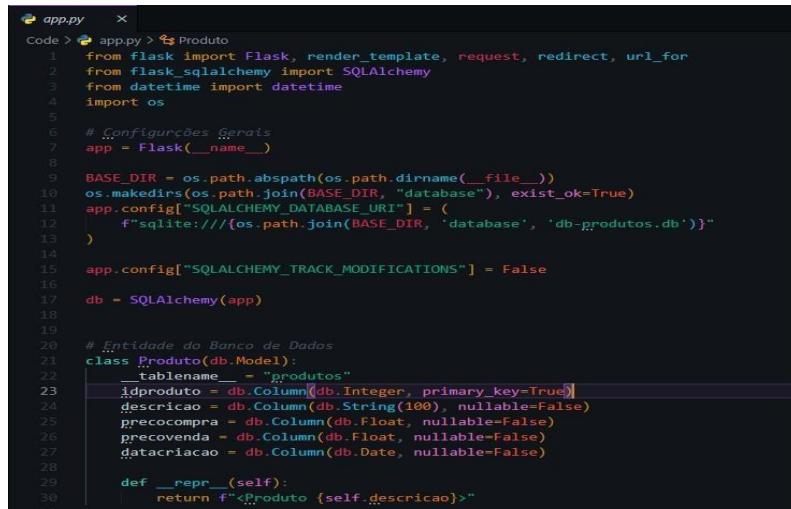
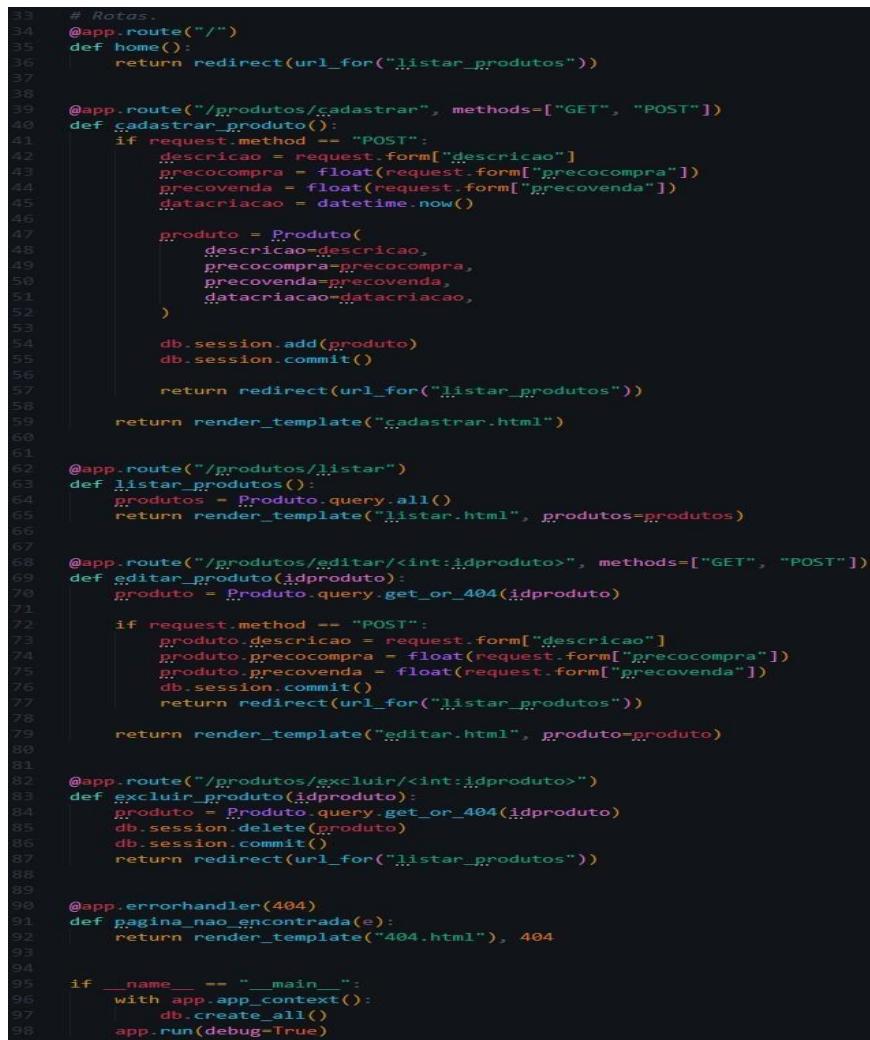


## app.py -



```
app.py  x
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.



```
1  # Rotas:
2  @app.route("/")
3  def home():
4      return redirect(url_for("listar_produtos"))
5
6
7  @app.route("/produtos/cadastrar", methods=["GET", "POST"])
8  def cadastrar_produto():
9      if request.method == "POST":
10          descricao = request.form["descricao"]
11          precocompra = float(request.form["precocompra"])
12          precovenda = float(request.form["precovenda"])
13          datacriacao = datetime.now()
14
15          produto = Produto(
16              descricao=descricao,
17              precocompra=precocompra,
18              precovenda=precovenda,
19              datacriacao=datacriacao,
20          )
21
22          db.session.add(produto)
23          db.session.commit()
24
25          return redirect(url_for("listar_produtos"))
26
27      return render_template("cadastrar.html")
28
29
30  @app.route("/produtos/listar")
31  def listar_produtos():
32      produtos = Produto.query.all()
33      return render_template("listar.html", produtos=produtos)
34
35
36  @app.route("/produtos/editar<int:idproduto>", methods=["GET", "POST"])
37  def editar_produto(idproduto):
38      produto = Produto.query.get_or_404(idproduto)
39
40      if request.method == "POST":
41          produto.descricao = request.form["descricao"]
42          produto.precocompra = float(request.form["precocompra"])
43          produto.precovenda = float(request.form["precovenda"])
44          db.session.commit()
45          return redirect(url_for("listar_produtos"))
46
47      return render_template("editar.html", produto=produto)
48
49
50  @app.route("/produtos/excluir<int:idproduto>")
51  def excluir_produto(idproduto):
52      produto = Produto.query.get_or_404(idproduto)
53      db.session.delete(produto)
54      db.session.commit()
55      return redirect(url_for("listar_produtos"))
56
57
58  @app.errorhandler(404)
59  def pagina_nao_encontrada(e):
60      return render_template("404.html"), 404
61
62
63  if __name__ == "__main__":
64      with app.app_context():
65          db.create_all()
66      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/ visualização da tabela de produtos e valores

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

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

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