Jordy Araújo







```
1 user@PC:~$ mkdir projeto
 2 user@PC:~$ cd projeto
 3 user@PC:~/projeto$ python3 -m venv .venv
 4 user@PC:~/projeto$ source .venv/bin/activate
 5 (.venv) user@PC:~/projeto$ pip install flask
 6 Collecting flask
 7 Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
 8 Collecting importlib-metadata>=3.6.0; python_version <
   "3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
   (21 \text{ kB})
10 Collecting Werkzeug>=2.2.2
```

```
1 user@PC:~$ mkdir projeto
 2 user@PC:~$ cd projeto
 3 user@PC:~/projeto$ python3 -m venv .venv
 4 user@PC:~/projeto$ source .venv/bin/activate
 5 (.venv) user@PC:~/projeto$ pip install flask
 6 Collecting flask
 7 Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
 8 Collecting importlib-metadata>=3.6.0; python_version <
   "3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
   (21 \text{ kB})
10 Collecting Werkzeug>=2.2.2
```

```
1 user@PC:~$ mkdir projeto
 2 user@PC:~$ cd projeto
 3 user@PC:~/projeto$ python3 -m venv .venv
 4 user@PC:~/projeto$ source .venv/bin/activate
 5 (.venv) user@PC:~/projeto$ pip install flask
 6 Collecting flask
 7 Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
 8 Collecting importlib-metadata>=3.6.0; python_version <
   "3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
   (21 \text{ kB})
10 Collecting Werkzeug>=2.2.2
```

```
1 user@PC:~$ mkdir projeto
 2 user@PC:~$ cd projeto
 3 user@PC:~/projeto$ python3 -m venv .venv
 4 user@PC:~/projeto$ source .venv/bin/activate
 5 (.venv) user@PC:~/projeto$ pip install flask
 6 Collecting flask
 7 Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
 8 Collecting importlib-metadata>=3.6.0; python_version <
   "3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
   (21 \text{ kB})
10 Collecting Werkzeug>=2.2.2
```

```
1 user@PC:~$ mkdir projeto
 2 user@PC:~$ cd projeto
 3 user@PC:~/projeto$ python3 -m venv .venv
 4 user@PC:~/projeto$ source .venv/bin/activate
 5 (.venv) user@PC:~/projeto$ pip install flask
 6 Collecting flask
 7 Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
 8 Collecting importlib-metadata>=3.6.0; python_version <
   "3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
   (21 \text{ kB})
10 Collecting Werkzeug>=2.2.2
```

```
P Collecting flask
     Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
 8 Collecting importlib-metadata>=3.6.0; python_version <</pre>
   "3.10"
    Using cached importlib_metadata-4.12.0-py3-none-any.whl
   (21 \text{ kB})
   Collecting Werkzeug>=2.2.2
11
     Using cached Werkzeug-2.2.2-py3-none-any.whl (232 kB)
12 Collecting itsdangerous>=2.0
     Using cached itsdangerous-2.1.2-py3-none-any.whl (15
13
   kB)
   Collecting Jinja2>=3.0
     Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
15
```

```
18 Collecting click>=8.0
Using cached click-8.1.3-py3-none-any.whl (96 kB)
   Collecting zipp > = 0.5
     Using cached zipp-3.8.1-py3-none-any.whl (5.6 kB)
   Collecting MarkupSafe>=2.1.1
     Using cached MarkupSafe-2.1.1-cp38-cp38-
   manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
22 Installing collected packages: zipp, importlib-metadata,
   MarkupSafe, Werkzeug, itsdangerous, Jinja2, click, flask
23 Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1
   Werkzeug-2.2.2 click-8.1.3 flask-2.2.2 importlib-
   metadata-4.12.0 itsdangerous-2.1.2 zipp-3.8.1
24 (.venv) user@PC:~/projeto$ pip freeze > requirements.txt
```

```
18 Collecting click>=8.0
Using cached click-8.1.3-py3-none-any.whl (96 kB)
   Collecting zipp > = 0.5
     Using cached zipp-3.8.1-py3-none-any.whl (5.6 kB)
   Collecting MarkupSafe>=2.1.1
     Using cached MarkupSafe-2.1.1-cp38-cp38-
   manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
22 Installing collected packages: zipp, importlib-metadata,
   MarkupSafe, Werkzeug, itsdangerous, Jinja2, click, flask
23 Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1
   Werkzeug-2.2.2 click-8.1.3 flask-2.2.2 importlib-
   metadata-4.12.0 itsdangerous-2.1.2 zipp-3.8.1
24 (.venv) user@PC:~/projeto$ pip freeze > requirements.txt
```

```
from flask import Flask
   def create_app():
     app = Flask(__name___)
5
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
10
     @app.route('/ola/<nome>')
11
     def ola_nome(nome):
12
       return f'Olá, {nome}!'
13
15
     return app
```

```
from flask import Flask
   def create_app():
     app = Flask(__name___)
5
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
10
     @app.route('/ola/<nome>')
11
     def ola_nome(nome):
12
       return f'Olá, {nome}!'
13
15
     return app
```

```
from flask import Flask
   def create_app():
     app = Flask(__name___)
5
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
10
     @app.route('/ola/<nome>')
11
     def ola_nome(nome):
12
       return f'Olá, {nome}!'
13
15
     return app
```

```
from flask import Flask
 2
 3
   def create_app():
     app = Flask(__name___)
 5
 6
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
10
     @app.route('/ola/<nome>')
11
     def ola_nome(nome):
12
       return f'Olá, {nome}!'
13
15
     return app
```

```
from flask import Flask
   def create_app():
     app = Flask(__name___)
 5
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
10
     @app.route('/ola/<nome>')
11
12
     def ola_nome(nome):
       return f'Olá, {nome}!'
13
15
     return app
```

```
from flask import Flask
 2
 3
   def create_app():
     app = Flask(__name___)
 5
 6
     @app.route('/ola')
     def ola_mundo():
        return '<h1>0lá, mundo!</h1>'
10
     @app.route('/ola/<nome>')
11
     def ola_nome(nome):
12
        return f'Olá, {nome}!'
13
14
15
     return app
```

```
from flask import Flask, render_template
   def create_app():
     app = Flask(__name___)
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
10
     @app.route('/ola/<nome>')
11
     def ola_nome(nome):
12
13
       return f'Olá, {nome}!'
15
     @app.route('/cadastrar')
     def form_cadastrar():
16
       return render_template('cadastrar.html')
```

```
def create_app():
     app = Flask(__name___)
 6
     @app.route('/ola')
     def ola_mundo():
 8
10
11
     @app.route('/ola/<nome>')
     def ola_nome(nome):
12
       return f'Olá, {nome}!'
13
     @app.route('/cadastrar')
15
     def form_cadastrar():
16
        return render_template('cadastrar.html')
17
```

```
from flask import Flask, render_template
   def create_app():
     app = Flask(__name___)
 6
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
10
11
     @app.route('/ola/<nome>')
     def ola_nome(nome):
12
13
       return f'Olá, {nome}!'
15
     @app.route('/cadastrar')
16
     def form_cadastrar():
       return render_template('cadastrar.html')
17
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Cadastrar</title>
   </head>
   <body>
     <form action="/cadastrar" method="post">
10
       <input type="text" name="nome" id="">
11
       <button type="submit">Enviar</button>
12
     </form>
13
14 </body>
15 </html>
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Cadastrar</title>
   </head>
   <body>
     <form action="/cadastrar" method="post">
10
       <input type="text" name="nome" id="">
11
       <button type="submit">Enviar/button>
12
   </form>
14 </body>
15 </html>
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Cadastrar</title>
   </head>
   <body>
     <form action="/cadastrar" method="post">
10
       <input type="text" name="nome" id="">
       <button type="submit">Enviar
12
     </form>
14 </body>
15 </html>
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Cadastrar</title>
  </head>
   <body>
     <form action="/cadastrar" method="post">
10
       <input type="text" name="nome" id="">
       <button type="submit">Enviar
   </form>
14 </body>
15 </html>
```

```
import sqlite3
   import click
   from flask import current_app, g
5
   def get_db():
    if 'db' not in g:
       g.db = sqlite3.connect(
10
         current_app.config['DATABASE'],
         detect_types=sqlite3.PARSE_DECLTYPES
11
12
       g.db.row_factory = sqlite3.Row
     return g.db
16
```

```
import sqlite3
   import click
   from flask import current_app, g
   def get_db():
    if 'db' not in g:
       g.db = sqlite3.connect(
10
         current_app.config['DATABASE'],
         detect_types=sqlite3.PARSE_DECLTYPES
11
       g.db.row_factory = sqlite3.Row
     return g.db
16
```

```
import sqlite3
   import click
   from flask import current_app, g
5
6
   def get_db():
   if 'db' not in g:
       g.db = sqlite3.connect(
10
         current_app.config['DATABASE'],
         detect_types=sqlite3.PARSE_DECLTYPES
11
12
       g.db.row_factory = sqlite3.Row
     return g.db
16
```

```
import click
   from flask import current_app, g
 6
   def get_db():
     if 'db' not in g:
 8
       g.db = sqlite3.connect(
         current_app.config['DATABASE'],
10
11
         detect_types=sqlite3.PARSE_DECLTYPES
12
13
       g.db.row_factory = sqlite3.Row
14
15
     return g.db
16
18 def close_db(e=None):
```

```
g.db.row_factory = sqlite3.Row
     return g.db
15
16
   def close_db(e=None):
18
     db = g.pop('db', None)
19
20
     if db is not None:
21
       db.close()
22
   def init_db():
     db = get_db()
26
```

```
def close_db(e=None):
     db = g.pop('db', None)
19
20
     if db is not None:
22
       db.close()
25
   def init_db():
     db = get_db()
26
27
     with current_app.open_resource('schema.sql') a
28
29
       db.executescript(f.read().decode('utf8'))
30
31
   @click.command('init-db')
   def init_db_command():
     init db()
```

```
def init_db():
26
     db = get_db()
     with current_app.open_resource('schema.sql')
28
       db.executescript(f.read().decode('utf8'))
29
30
31
   @click.command('init-db')
   def init_db_command():
33
     init_db()
34
     click.echo('Banco de dados inicializado.')
35
36
   def init_app(app):
39
     app.teardown appcontext(close db)
```

```
def init_db():
26
     db = get_db()
     with current_app.open_resource('schema.sql')
28
       db.executescript(f.read().decode('utf8'))
29
30
31
   @click.command('init-db')
   def init_db_command():
     init_db()
     click.echo('Banco de dados inicializado.')
35
36
   def init_app(app):
39
     app.teardown appcontext(close db)
```

```
import sqlite3
 2
   import click
   from flask import current_app, g
 5
 6
   def get_db():
     if 'db' not in g:
 8
       g.db = sqlite3.connect(
10
         current_app.config['DATABASE'],
         detect_types=sqlite3.PARSE_DECLTYPES
11
12
       g.db.row_factory = sqlite3.Row
13
14
     return g.db
15
16
17
```

```
DROP TABLE IF EXISTS pessoa;
3 CREATE TABLE pessoa (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    nome TEXT UNIQUE NOT NULL
 );
```

```
import os
   import db
   from flask import Flask, render_template, reques
6
   def create_app():
     app = Flask(__name___)
10
       os.makedirs(app.instance_path)
11
     except OSError:
       pass
13
     app.config.from_mapping(
       DATABASE=os.path.join(app.instance_path, f'
16
```

```
import
   from flask import Flask, render_template, reques
   def create_app():
 8
     app = Flask(__name___)
 9
     try:
10
       os.makedirs(app.instance_path)
     except OSError:
11
12
       pass
13
     app.config.from_mapping(
       DATABASE=os.path.join(app.instance_path, f'
16
     db.init_app(app)
18
```

```
def create_app():
     app = Flask(__name___)
       os.makedirs(app.instance_path)
10
11
     except OSError:
       pass
13
     app.config.from_mapping(
14
       DATABASE=os.path.join(app.instance_path, f'
15
16
     db.init_app(app)
18
19
20
     @app.route('/ola')
     def ola_mundo():
       return '<h1>0lá, mundo!</h1>'
```

```
import os
   import db
   from flask import Flask, render_template, reques
6
   def create_app():
     app = Flask(__name___)
10
       os.makedirs(app.instance_path)
11
     except OSError:
       pass
13
     app.config.from_mapping(
       DATABASE=os.path.join(app.instance_path, f'
16
```

```
10
       os.makedirs(app.instance_path)
11
     except OSError:
12
       pass
     app.config.from_mapping(
       DATABASE=os.path.join(app.instance_path, f'
15
16
     db.init_app(app)
18
19
     @app.route('/ola')
20
     def ola_mundo():
21
       return '<h1>0lá, mundo!</h1>'
     @app.route('/ola/<nome>')
     def ola_nome(nome):
25
```

```
import os
   import db
 3
   from flask import Flask, render_template, reques
 5
   def create_app():
     app = Flask(__name___)
     try:
10
       os.makedirs(app.instance_path)
11
     except OSError:
12
       pass
13
     app.config.from_mapping(
14
       DATABASE=os.path.join(app.instance_path, f'
15
16
17
```

```
import os
   import db
   from flask import Flask, render_template, reques
6
   def create_app():
     app = Flask(__name___)
 9
10
       os.makedirs(app.instance_path)
11
12
     except OSError:
       pass
     app.config.from_mapping(
       DATABASE=os.path.join(app.instance_path, f'
16
```

```
25
     @app.route('/ola/<nome>')
26
     def ola_nome(nome):
       return f'Olá, {nome}!'
28
     @app.route('/cadastrar')
29
30
     def form_cadastrar():
31
       return render_template('cadastrar.html')
32
33
     @app.route('/cadastrar', methods=['POST'])
     def cadastrar():
34
35
       nome = request.form['nome']
36
       banco = db.get_db()
       banco.execute(
38
         [nome],
39
40
       banco.commit()
```

```
26
     def ola_nome(nome):
       return f'Olá, {nome}!'
28
29
     @app.route('/cadastrar')
     def form_cadastrar():
30
       return render_template('cadastrar.html')
31
32
33
     @app.route('/cadastrar', methods=['POST'])
     def cadastrar():
       nome = request.form['nome']
35
       banco = db.get_db()
36
       banco.execute(
38
39
         [nome],
40
       banco.commit()
42
```

```
return f'Olá, {nome}!'
28
29
     @app.route('/cadastrar')
     def form_cadastrar():
30
       return render_template('cadastrar.html')
32
33
     @app.route('/cadastrar', methods=['POST'])
     def cadastrar():
       nome = request.form['nome']
35
       banco = db.get_db()
36
       banco.execute(
38
39
         [nome],
40
41
       banco.commit()
42
```

```
28
29
     @app.route('/cadastrar')
     def form_cadastrar():
30
31
       return render_template('cadastrar.html')
32
33
     @app.route('/cadastrar', methods=['POST'])
     def cadastrar():
       nome = request.form['nome']
35
       banco = db.get_db()
36
37
       banco.execute(
          "INSERT INTO pessoa (nome) VALUES (?)",
38
39
         [nome],
40
41
       banco.commit()
42
       return f'{nome} cadastrado(a) com sucesso!'
```

```
28
29
     @app.route('/cadastrar')
     def form_cadastrar():
30
31
       return render_template('cadastrar.html')
32
33
     @app.route('/cadastrar', methods=['POST'])
     def cadastrar():
       nome = request.form['nome']
35
       banco = db.get_db()
36
       banco.execute(
38
39
         [nome],
40
41
       banco.commit()
42
       return f'{nome} cadastrado(a) com sucesso!'
```

```
28
29
     @app.route('/cadastrar')
     def form_cadastrar():
30
31
       return render_template('cadastrar.html')
32
33
     @app.route('/cadastrar', methods=['POST'])
     def cadastrar():
       nome = request.form['nome']
35
       banco = db.get_db()
36
       banco.execute(
38
39
         [nome],
40
41
       banco.commit()
42
       return f'{nome} cadastrado(a) com sucesso!'
43
```

```
28
29
     @app.route('/cadastrar')
     def form_cadastrar():
30
31
       return render_template('cadastrar.html')
32
33
     @app.route('/cadastrar', methods=['POST'])
     def cadastrar():
34
       nome = request.form['nome']
35
       banco = db.get_db()
36
       banco.execute(
37
         "INSERT INTO pessoa (nome) VALUES (?)",
38
39
         [nome],
40
41
       banco.commit()
42
       return f'{nome} cadastrado(a) com sucesso!'
43
```

```
import os
   import db
   from flask import Flask, render_template, reques
6
   def create_app():
     app = Flask(__name___)
 9
10
       os.makedirs(app.instance_path)
11
12
     except OSError:
       pass
     app.config.from_mapping(
16
       DATABASE=os.path.join(app.instance_path, f'
```

```
36
       banco = db.get_db()
       banco.execute(
38
39
         [nome],
40
41
       banco.commit()
42
43
       return f'Olá, {nome}!'
45
     @app.route('/listar')
     def listar():
46
       banco = db.get_db()
48
       pessoas = banco.execute(
          'SELECT * FROM pessoa ORDER BY id DESC'
49
       ).fetchall()
50
       return render_template('listar.html', pessoa
51
```

```
36
       banco = db.get_db()
       banco.execute(
38
39
         [nome],
40
41
       banco.commit()
42
43
       return f'Olá, {nome}!'
45
     @app.route('/listar')
     def listar():
46
       banco = db.get_db()
48
       pessoas = banco.execute(
          'SELECT * FROM pessoa ORDER BY id DESC'
50
       ).fetchall()
       return render_template('listar.html', pessoa
51
```

```
36
       banco = db.get_db()
       banco.execute(
38
39
         [nome],
40
41
       banco.commit()
42
43
       return f'Olá, {nome}!'
45
     @app.route('/listar')
     def listar():
46
       banco = db.get_db()
47
48
       pessoas = banco.execute(
          'SELECT * FROM pessoa ORDER BY id DESC'
49
       ).fetchall()
50
       return render_template('listar.html', pessoa
51
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Listar</title>
   </head>
   <body>
10
     <l
       {% for pessoa in pessoas %}
12
13
          {{ pessoa['nome'] }}
    {% endfor %}
16
     </body>
```

```
<html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Listar</title>
   </head>
   <body>
     ul>
10
11
       {% for pessoa in pessoas %}
         <
12
13
           {{ pessoa['nome'] }}
14
         {% endfor %}
15
16
     17 </body>
```

```
<html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Listar</title>
   </head>
   <body>
     <l
10
       {% for pessoa in pessoas %}
11
         <
12
13
           {{ pessoa['nome'] }}
14
         {% endfor %}
15
16
     17 </body>
```

```
<html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Listar</title>
   </head>
   <body>
     <l
10
       {% for pessoa in pessoas %}
         <
12
13
           {{ pessoa['nome'] }}
14
         {% endfor %}
16
     17 </body>
```

```
<html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Listar</title>
   </head>
   <body>
     ul>
10
11
       {% for pessoa in pessoas %}
         <
12
13
           {{ pessoa['nome'] }}
14
         {% endfor %}
15
16
     17 </body>
```

```
import os
   import db
   from flask import Flask, render_template, reques
6
   def create_app():
     app = Flask(__name___)
 9
10
       os.makedirs(app.instance_path)
11
12
     except OSError:
       pass
     app.config.from_mapping(
16
       DATABASE=os.path.join(app.instance_path, f'
```

```
def cadastrar():
35
       nome = request.form['nome']
36
       banco = db.get_db()
       banco.execute(
38
39
         [nome],
40
41
       banco.commit()
42
       return redirect("listar")
43
45
     @app.route('/listar')
46
     def listar():
       banco = db.get_db()
48
       pessoas = banco.execute(
          'SELECT * FROM pessoa ORDER BY id DESC'
49
```

```
import os
   import db
   from flask import Flask, render_template, reques
6
   def create_app():
     app = Flask(__name___)
 9
10
       os.makedirs(app.instance_path)
11
12
     except OSError:
       pass
     app.config.from_mapping(
       DATABASE=os.path.join(app.instance_path, f'
16
```

```
44
45
     @app.route('/listar')
46
     def listar():
       banco = db.get_db()
       pessoas = banco.execute(
48
          'SELECT * FROM pessoa ORDER BY id DESC'
50
       ).fetchall()
       return render_template('listar.html', pessoa
52
53
     @app.route('/<id>/editar')
     def form_editar(id):
54
       banco = db.get_db()
55
       pessoa = banco.execute(
56
57
          'SELECT * FROM pessoa WHERE id = (?)',
58
         [id]
       ).fetchone()
       return render_template('editar.html', pessoa
60
```

```
44
45
     @app.route('/listar')
46
     def listar():
       banco = db.get_db()
       pessoas = banco.execute(
48
          'SELECT * FROM pessoa ORDER BY id DESC'
50
       ).fetchall()
       return render_template('listar.html', pessoa
52
53
     @app.route('/<id>/editar')
     def form_editar(id):
54
55
       banco = db.get_db()
56
       pessoa = banco.execute(
57
          'SELECT * FROM pessoa WHERE id = (?)',
          [id]
58
59
       ).fetchone()
       return render_template('editar.html', pessoa
60
```

```
44
45
     @app.route('/listar')
46
     def listar():
       banco = db.get_db()
       pessoas = banco.execute(
48
          'SELECT * FROM pessoa ORDER BY id DESC'
50
       ).fetchall()
       return render_template('listar.html', pessoa
52
53
     @app.route('/<id>/editar')
     def form_editar(id):
54
       banco = db.get_db()
55
       pessoa = banco.execute(
56
          'SELECT * FROM pessoa WHERE id = (?)',
         [id]
58
       ).fetchone()
       return render_template('editar.html', pessoa
60
```

```
44
45
     @app.route('/listar')
46
     def listar():
       banco = db.get_db()
       pessoas = banco.execute(
48
          'SELECT * FROM pessoa ORDER BY id DESC'
50
       ).fetchall()
       return render_template('listar.html', pessoa
52
53
     @app.route('/<id>/editar')
     def form_editar(id):
54
55
       banco = db.get_db()
       pessoa = banco.execute(
56
57
          'SELECT * FROM pessoa WHERE id = (?)',
         [id]
58
59
       ).fetchone()
       return render_template('editar.html', pessoa
60
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Editar</title>
   </head>
   <body>
10
     <form action="/atualizar" method="post">
       <input type="hidden" name="id" value="{{ pes</pre>
11
       <label for="nome">Insira um novo nome para
12
13
       <input type="text" name="nome">
14
       <button type="submit">Enviar/button>
15
     </form>
   </body>
17 </html>
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Editar</title>
   </head>
   <body>
10
     <form action="/atualizar" method="post">
       <input type="hidden" name="id" value="{{ pes</pre>
11
       <label for="nome">Insira um novo nome para
13
       <input type="text" name="nome">
       <button type="submit">Enviar</button>
15
     </form>
   </body>
17 </html>
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Editar</title>
   </head>
   <body>
10
     <form action="/atualizar" method="post">
       <input type="hidden" name="id" value="{{ pes</pre>
       <label for="nome">Insira um novo nome para {
12
13
       <input type="text" name="nome">
       <button type="submit">Enviar</button>
15
     </form>
   </body>
17 </html>
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Editar</title>
   </head>
   <body>
10
     <form action="/atualizar" method="post">
       <input type="hidden" name="id" value="{{ pes</pre>
       <label for="nome">Insira um novo nome para
13
       <input type="text" name="nome">
       <button type="submit">Enviar/button>
14
15
     </form>
   </body>
   </html>
```

```
1 <!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Editar</title>
   </head>
   <body>
10
     <form action="/atualizar" method="post">
       <input type="hidden" name="id" value="{{ pes</pre>
       <label for="nome">Insira um novo nome para
12
13
       <input type="text" name="nome">
       <button type="submit">Enviar</button>
15
     </form>
   </body>
17 </html>
```

```
import os
   import db
   from flask import Flask, render_template, reques
5
6
   def create_app():
     app = Flask(__name___)
8
9
10
11
       os.makedirs(app.instance_path)
     except OSError:
       pass
     app.config.from_mapping(
15
       DATABASE=os.path.join(app.instance_path, f'
```

```
def form_editar(id):
       banco = db.get_db()
55
       pessoa = banco.execute(
56
         'SELECT * FROM pessoa WHERE id = (?)',
57
58
         [id]
59
       ).fetchone()
60
       return render_template('editar.html', pessoa
61
62
     @app.route('/atualizar', methods=['POST'])
63
     def atualizar():
       id = request.form['id']
64
       nome = request.form['nome']
65
66
       banco = db.get_db()
67
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
         (nome, id)
69
```

```
55
       banco = db.get_db()
       pessoa = banco.execute(
56
         'SELECT * FROM pessoa WHERE id = (?)',
57
58
         [id]
59
       ).fetchone()
       return render_template('editar.html', pessoa
60
61
62
     @app.route('/atualizar', methods=['POST'])
     def atualizar():
63
64
       id = request.form['id']
       nome = request.form['nome']
65
       banco = db.get_db()
66
       banco.execute(
67
68
         'UPDATE pessoa SET nome = ? WHERE id = ?'
         (nome, id)
69
70
```

```
pessoa = banco.execute(
          'SELECT * FROM pessoa WHERE id = (?)',
58
         [id]
59
       ).fetchone()
       return render_template('editar.html', pessoa
60
61
62
     @app.route('/atualizar', methods=['POST'])
63
     def atualizar():
64
       id = request.form['id']
65
       nome = request.form['nome']
       banco = db.get_db()
66
67
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
          (nome, id)
69
70
       banco.commit()
```

```
pessoa = banco.execute(
          'SELECT * FROM pessoa WHERE id = (?)',
58
         [id]
       ).fetchone()
59
60
       return render_template('editar.html', pessoa
61
62
     @app.route('/atualizar', methods=['POST'])
63
     def atualizar():
64
       id = request.form['id']
65
       nome = request.form['nome']
       banco = db.get_db()
66
67
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
         (nome, id)
69
71
       banco.commit()
```

```
pessoa = banco.execute(
          'SELECT * FROM pessoa WHERE id = (?)',
58
         [id]
59
       ).fetchone()
       return render_template('editar.html', pessoa
60
61
62
     @app.route('/atualizar', methods=['POST'])
63
     def atualizar():
64
       id = request.form['id']
65
       nome = request.form['nome']
       banco = db.get_db()
66
67
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
         (nome, id)
69
       banco.commit()
```

```
pessoa = banco.execute(
          'SELECT * FROM pessoa WHERE id = (?)',
58
         [id]
59
       ).fetchone()
       return render_template('editar.html', pessoa
60
61
     @app.route('/atualizar', methods=['POST'])
62
     def atualizar():
63
64
       id = request.form['id']
65
       nome = request.form['nome']
       banco = db.get_db()
66
67
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
          (nome, id)
69
70
71
       banco.commit()
```

```
<!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Listar</title>
   </head>
   <body>
10
     <l
       {% for pessoa in pessoas %}
12
           {{ pessoa['nome'] }}
14
           - <a href="/{{ pessoa['id'] }}/editar">E
         15
16
       {% endfor %}
```

```
import os
   import db
   from flask import Flask, render_template, reques
5
6
   def create_app():
     app = Flask(__name___)
8
9
10
11
       os.makedirs(app.instance_path)
     except OSError:
       pass
     app.config.from_mapping(
15
       DATABASE=os.path.join(app.instance_path, f'
```

```
nome = request.form['nome']
65
       banco = db.get_db()
66
67
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
         (nome, id)
69
       banco.commit()
       return redirect("listar")
     @app.route('/<id>/deletar')
     def deletar(id):
75
       banco = db.get_db()
76
       pessoa = banco.execute(
         'DELETE FROM pessoa WHERE id = (?)',
         [id]
80
```

```
66
       banco = db.get_db()
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
          (nome, id)
69
       banco.commit()
       return redirect("listar")
     @app.route('/<id>/deletar')
75
     def deletar(id):
76
       banco = db.get_db()
       pessoa = banco.execute(
77
          'DELETE FROM pessoa WHERE id = (?)',
78
79
          [id]
80
81
       banco.commit()
```

```
66
       banco = db.get_db()
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
         (nome, id)
69
       banco.commit()
       return redirect("listar")
     @app.route('/<id>/deletar')
     def deletar(id):
76
       banco = db.get_db()
       pessoa = banco.execute(
          'DELETE FROM pessoa WHERE id = (?)',
78
         [id]
80
       banco.commit()
81
```

```
66
       banco = db.get_db()
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
         (nome, id)
69
70
       banco.commit()
       return redirect("listar")
     @app.route('/<id>/deletar')
     def deletar(id):
76
       banco = db.get_db()
       pessoa = banco.execute(
          'DELETE FROM pessoa WHERE id = (?)',
78
         [id]
80
81
       banco.commit()
```

```
banco = db.get_db()
66
       banco.execute(
         'UPDATE pessoa SET nome = ? WHERE id = ?'
68
          (nome, id)
69
70
       banco.commit()
       return redirect("listar")
74
     @app.route('/<id>/deletar')
75
     def deletar(id):
76
       banco = db.get_db()
       pessoa = banco.execute(
77
          'DELETE FROM pessoa WHERE id = (?)',
78
79
          [id]
80
81
       banco.commit()
```

```
<!DOCTYPE html>
   <html lang="pt-br">
   <head>
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE</pre>
     <meta name="viewport" content="width=device-wi</pre>
     <title>Listar</title>
   </head>
   <body>
10
     <l
       {% for pessoa in pessoas %}
           {{ pessoa['nome'] }}
           - <a href="/{{ pessoa['id'] }}/editar">E
           - <a href="/{{ pessoa['id'] }}/deletar">
15
16
         {% endfor %}
```





linkedin.com/in/jordyaraujo

jordyaraujo.github.io