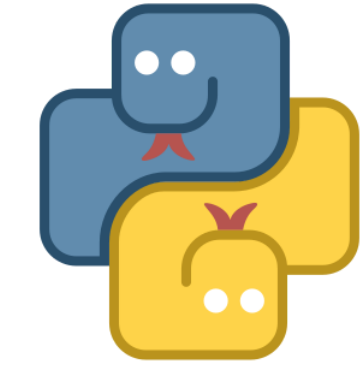
Jordy Araújo



grupy.rn

# Flask

Flask

web development,
one drop at a time

# Flask

```
user@PC:~$ mkdir projeto
user@PC:~$ cd projeto
user@PC:~/projeto$ python3 -m venv .venv
user@PC:~/projeto$ source .venv/bin/activate
(.venv) user@PC:~/projeto$ pip install flask
Collecting flask
  Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
Collecting importlib-metadata>=3.6.0; python_version <
"3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
(21 kB)
Collecting Werkzeug>=2.2.2
```

(sempre crie uma venv)

# Flask

```
user@PC:~$ mkdir projeto
user@PC:~$ cd projeto
user@PC:~/projeto$ python3 -m venv .venv
user@PC:~/projeto$ source .venv/bin/activate
(.venv) user@PC:~/projeto$ pip install flask
Collecting flask
  Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
Collecting importlib-metadata>=3.6.0; python_version <
"3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
(21 kB)
Collecting Werkzeug>=2.2.2
```

(sempre crie uma venv)

# Flask

```
user@PC:~$ mkdir projeto
user@PC:~$ cd projeto
user@PC:~/projeto$ python3 -m venv .venv
user@PC:~/projeto$ source .venv/bin/activate
(.venv) user@PC:~/projeto$ pip install flask
Collecting flask
  Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
Collecting importlib-metadata>=3.6.0; python_version <
"3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
(21 kB)
Collecting Werkzeug>=2.2.2
```

(sempre crie uma venv)

# Flask

```
user@PC:~$ mkdir projeto
user@PC:~$ cd projeto
user@PC:~/projeto$ python3 -m venv .venv
user@PC:~/projeto$ source .venv/bin/activate
(.venv) user@PC:~/projeto$ pip install flask
Collecting flask
  Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
Collecting importlib-metadata>=3.6.0; python_version <
"3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
(21 kB)
Collecting Werkzeug>=2.2.2
```

(sempre crie uma venv)

# Flask

```
user@PC:~$ mkdir projeto
user@PC:~$ cd projeto
user@PC:~/projeto$ python3 -m venv .venv
user@PC:~/projeto$ source .venv/bin/activate
(.venv) user@PC:~/projeto$ pip install flask
Collecting flask
  Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
Collecting importlib-metadata>=3.6.0; python_version <
"3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl
(21 kB)
Collecting Werkzeug>=2.2.2
```

(sempre crie uma venv)

# Flask

```
(.venv) user@PC:~/projeto$ pip install flask
Collecting flask
  Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
Collecting importlib-metadata>=3.6.0; python_version < "3.10"
  Using cached importlib_metadata-4.12.0-py3-none-any.whl (21 kB)
Collecting Werkzeug>=2.2.2
  Using cached Werkzeug-2.2.2-py3-none-any.whl (232 kB)
Collecting itsdangerous>=2.0
  Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Jinja2>=3.0
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
```

(sempre crie uma venv)

# Flask

```
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)
Installing collected packages: zipp, importlib-metadata,
  MarkupSafe, Werkzeug, itsdangerous, Jinja2, click, flask
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
(.venv) user@PC:~/projeto$ pip freeze > requirements.txt
```

(sempre crie uma venv)

# Flask

```
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)
Installing collected packages: zipp, importlib-metadata,
MarkupSafe, Werkzeug, itsdangerous, Jinja2, click, flask
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
(.venv) user@PC:~/projeto$ pip freeze > requirements.txt
```

(sempre crie uma venv)

# O mínimo do mínimo

```python
from flask import Flask


def create_app():
  app = Flask(__name__)

  @app.route('/ola')
  def ola_mundo():
    return '<h1>Olá, mundo!</h1>'

  @app.route('/ola/<nome>')
  def ola_nome(nome):
    return f'Olá, {nome}!'

  return app
```

# O mínimo do mínimo

```python
from flask import Flask


def create_app():
    app = Flask(__name__)

    @app.route('/ola')
    def ola_mundo():
        return '<h1>Olá, mundo!</h1>'

    @app.route('/ola/<nome>')
    def ola_nome(nome):
        return f'Olá, {nome}!'

    return app
```

# O mínimo do mínimo

```python
from flask import Flask


def create_app():
  app = Flask(__name__)

  @app.route('/ola')
  def ola_mundo():
    return '<h1>Olá, mundo!</h1>'

  @app.route('/ola/<nome>')
  def ola_nome(nome):
    return f'Olá, {nome}!'

  return app
```

# O mínimo do mínimo

```python
from flask import Flask


def create_app():
  app = Flask(__name__)

  @app.route('/ola')
  def ola_mundo():
    return '<h1>Olá, mundo!</h1>'

  @app.route('/ola/<nome>')
  def ola_nome(nome):
    return f'Olá, {nome}!'

  return app
```

# O mínimo do mínimo

```python
from flask import Flask


def create_app():
  app = Flask(__name__)

  @app.route('/ola')
  def ola_mundo():
    return '<h1>Olá, mundo!</h1>'

  @app.route('/ola/<nome>')
  def ola_nome(nome):
    return f'Olá, {nome}!'

  return app
```

# O mínimo do mínimo

```python
from flask import Flask


def create_app():
  app = Flask(__name__)

  @app.route('/ola')
  def ola_mundo():
    return '<h1>Olá, mundo!</h1>'

  @app.route('/ola/<nome>')
  def ola_nome(nome):
    return f'Olá, {nome}!'

  return app
```

# C.r.u.d.

```python
from flask import Flask, render_template


def create_app():
  app = Flask(__name__)

  @app.route('/ola')
  def ola_mundo():
    return '<h1>Olá, mundo!</h1>'

  @app.route('/ola/<nome>')
  def ola_nome(nome):
    return f'Olá, {nome}!'

  @app.route('/cadastrar')
  def form_cadastrar():
    return render_template('cadastrar.html')
```

# C.r.u.d.

```
 3
 4  def create_app():
 5      app = Flask(__name__)
 6
 7      @app.route('/ola')
 8      def ola_mundo():
 9          return '<h1>Olá, mundo!</h1>'
10
11      @app.route('/ola/<nome>')
12      def ola_nome(nome):
13          return f'Olá, {nome}!'
14
15      @app.route('/cadastrar')
16      def form_cadastrar():
17          return render_template('cadastrar.html')

18
```

# C.r.u.d.

```python
from flask import Flask, render_template


def create_app():
  app = Flask(__name__)

  @app.route('/ola')
  def ola_mundo():
    return '<h1>Olá, mundo!</h1>'

  @app.route('/ola/<nome>')
  def ola_nome(nome):
    return f'Olá, {nome}!'

  @app.route('/cadastrar')
  def form_cadastrar():
    return render_template('cadastrar.html')
```

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

```
import sqlite3

import click
from flask import current_app, g


def get_db():
  if 'db' not in g:
    g.db = sqlite3.connect(
      current_app.config['DATABASE'],
      detect_types=sqlite3.PARSE_DECLTYPES
    )
    g.db.row_factory = sqlite3.Row

  return g.db


```

# C.r.u.d.

```
import sqlite3

import click
from flask import current_app, g


def get_db():
  if 'db' not in g:
    g.db = sqlite3.connect(
      current_app.config['DATABASE'],
      detect_types=sqlite3.PARSE_DECLTYPES
    )
    g.db.row_factory = sqlite3.Row

  return g.db


```

# C.r.u.d.

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

18 def close_db(e=None):
```

# C.r.u.d.

```
12      )
13      g.db.row_factory = sqlite3.Row
14
15    return g.db
16
17
18  def close_db(e=None):
19    db = g.pop('db', None)
20
21    if db is not None:
22      db.close()
23
24
25  def init_db():
26    db = get_db()
27
```

# C.r.u.d.

```
18 def close_db(e=None):
19   db = g.pop('db', None)
20
21   if db is not None:
22     db.close()
23
24
25 def init_db():
26   db = get_db()
27
28   with current_app.open_resource('schema.sql') a
29     db.executescript(f.read().decode('utf8'))
30
31
32 @click.command('init-db')
33 def init_db_command():
34   init_db()
35   click.echo('Banco de dados inicializado.')
```

# C.r.u.d.

```
23
24
25 def init_db():
26   db = get_db()
27
28   with current_app.open_resource('schema.sql') a
29     db.executescript(f.read().decode('utf8'))
30
31
32 @click.command('init-db')
33 def init_db_command():
34   init_db()
35   click.echo('Banco de dados inicializado.')
36
37
38 def init_app(app):
39   app.teardown_appcontext(close_db)
```

# C.r.u.d.

```
23
24
25 def init_db():
26   db = get_db()
27
28   with current_app.open_resource('schema.sql') a
29     db.executescript(f.read().decode('utf8'))
30
31
32 @click.command('init-db')
33 def init_db_command():
34   init_db()
35   click.echo('Banco de dados inicializado.')
36
37
38 def init_app(app):
39   app.teardown_appcontext(close_db)
```

# C.r.u.d.

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

# C.r.u.d.

```sql
DROP TABLE IF EXISTS pessoa;

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

# C.r.u.d.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9   try:
10     os.makedirs(app.instance_path)
11   except OSError:
12     pass
13
14   app.config.from_mapping(
15     DATABASE=os.path.join(app.instance_path, f'{
16   )
17
```

# C.r.u.d.

```
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9   try:
10     os.makedirs(app.instance_path)
11   except OSError:
12     pass
13
14   app.config.from_mapping(
15     DATABASE=os.path.join(app.instance_path, f'{
16   )
17

18   db.init_app(app)
```

# C.r.u.d.

```
 7 def create_app():
 8   app = Flask(__name__)
 9   try:
10     os.makedirs(app.instance_path)
11   except OSError:
12     pass
13
14   app.config.from_mapping(
15     DATABASE=os.path.join(app.instance_path, f'{
16   )
17

18   db.init_app(app)
19
20   @app.route('/ola')
21   def ola_mundo():
22     return '<h1>Olá, mundo!</h1>'
```

# C.r.u.d.

```
import os
import db

from flask import Flask, render_template, reques


def create_app():
  app = Flask(__name__)
  try:
    os.makedirs(app.instance_path)
  except OSError:
    pass

  app.config.from_mapping(
    DATABASE=os.path.join(app.instance_path, f'{
  )

```

# C.r.u.d.

```
10      os.makedirs(app.instance_path)
11    except OSError:
12      pass
13
14    app.config.from_mapping(
15      DATABASE=os.path.join(app.instance_path, f'{
16    )
17
18    db.init_app(app)
19
20    @app.route('/ola')
21    def ola_mundo():
22      return '<h1>Olá, mundo!</h1>'
23
24    @app.route('/ola/<nome>')
25    def ola_nome(nome):
26      return f'Olá, {nome}!'
```

# C.r.u.d.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9   try:
10     os.makedirs(app.instance_path)
11   except OSError:
12     pass
13
14   app.config.from_mapping(
15     DATABASE=os.path.join(app.instance_path, f'{
16   )
17
```

# C.r.u.d.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9
10   try:
11     os.makedirs(app.instance_path)
12   except OSError:
13     pass
14
15   app.config.from_mapping(
16     DATABASE=os.path.join(app.instance_path, f'{
17   )
```

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# C.r.u.d.

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

# c.R.u.d.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9
10   try:
11     os.makedirs(app.instance_path)
12   except OSError:
13     pass
14
15   app.config.from_mapping(
16     DATABASE=os.path.join(app.instance_path, f'{
17   )
```

# c.R.u.d.

```
36      banco = db.get_db()
37      banco.execute(
38        "INSERT INTO pessoa (nome) VALUES (?)",
39        [nome],
40      )
41      banco.commit()
42
43      return f'Olá, {nome}!'
44
45    @app.route('/listar')
46    def listar():
47      banco = db.get_db()
48      pessoas = banco.execute(
49        'SELECT * FROM pessoa ORDER BY id DESC'
50      ).fetchall()
51      return render_template('listar.html', pessoa
```

# c.R.u.d.

```
36      banco = db.get_db()
37      banco.execute(
38        "INSERT INTO pessoa (nome) VALUES (?)",
39        [nome],
40      )
41      banco.commit()
42
43      return f'Olá, {nome}!'
44
45    @app.route('/listar')
46    def listar():
47      banco = db.get_db()
48      pessoas = banco.execute(
49        'SELECT * FROM pessoa ORDER BY id DESC'
50      ).fetchall()
51      return render_template('listar.html', pessoa
```

# c.R.u.d.

```
36      banco = db.get_db()
37      banco.execute(
38        "INSERT INTO pessoa (nome) VALUES (?)",
39        [nome],
40      )
41      banco.commit()
42
43      return f'Olá, {nome}!'
44
45    @app.route('/listar')
46    def listar():
47      banco = db.get_db()
48      pessoas = banco.execute(
49        'SELECT * FROM pessoa ORDER BY id DESC'
50      ).fetchall()
51      return render_template('listar.html', pessoa
```

# c.R.u.d.

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

# c.R.u.d.

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

# c.R.u.d.

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

# c.R.u.d.

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

# c.R.u.d.

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

# c.R.u.d.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9
10   try:
11     os.makedirs(app.instance_path)
12   except OSError:
13     pass
14
15   app.config.from_mapping(
16     DATABASE=os.path.join(app.instance_path, f'{
17   )
```

# c.R.u.d.

```
34    def cadastrar():
35      nome = request.form['nome']
36      banco = db.get_db()
37      banco.execute(
38        "INSERT INTO pessoa (nome) VALUES (?)",
39        [nome],
40      )
41      banco.commit()
42
43      return redirect("listar")
44
45    @app.route('/listar')
46    def listar():
47      banco = db.get_db()
48      pessoas = banco.execute(
49        'SELECT * FROM pessoa ORDER BY id DESC'
50      ).fetchall()
```

# c.r.U.d.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9
10   try:
11     os.makedirs(app.instance_path)
12   except OSError:
13     pass
14
15   app.config.from_mapping(
16     DATABASE=os.path.join(app.instance_path, f'{
17   )
```

# c.r.U.d.

```
44
45    @app.route('/listar')
46    def listar():
47      banco = db.get_db()
48      pessoas = banco.execute(
49        'SELECT * FROM pessoa ORDER BY id DESC'
50      ).fetchall()
51      return render_template('listar.html', pessoa
52
53    @app.route('/<id>/editar')
54    def form_editar(id):
55      banco = db.get_db()
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
60      return render_template('editar.html', pessoa
61
```

# c.r.U.d.

```
44
45    @app.route('/listar')
46    def listar():
47      banco = db.get_db()
48      pessoas = banco.execute(
49        'SELECT * FROM pessoa ORDER BY id DESC'
50      ).fetchall()
51      return render_template('listar.html', pessoa
52
53    @app.route('/<id>/editar')
54    def form_editar(id):
55      banco = db.get_db()
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
60      return render_template('editar.html', pessoa
61
```

# c.r.U.d.

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

# c.r.U.d.

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

# c.r.U.d.

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

# c.r.U.d.

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

# c.r.U.d.

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

# c.r.U.d.

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

# c.r.U.d.

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

# c.r.U.d.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9
10   try:
11     os.makedirs(app.instance_path)
12   except OSError:
13     pass
14
15   app.config.from_mapping(
16     DATABASE=os.path.join(app.instance_path, f'{
17   )
```

# c.r.U.d.

```
54    def form_editar(id):
55      banco = db.get_db()
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
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']
66      banco = db.get_db()
67      banco.execute(
68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
```

# c.r.U.d.

```
55      banco = db.get_db()
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
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']
66      banco = db.get_db()
67      banco.execute(
68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
```

# c.r.U.d.

```
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
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']
66      banco = db.get_db()
67      banco.execute(

68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
72      return redirect("listar")
```

# c.r.U.d.

```
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
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']
66      banco = db.get_db()
67      banco.execute(

68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
```

# c.r.U.d.

```
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
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']
66      banco = db.get_db()
67      banco.execute(

68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
72      return redirect("listar")
```

# c.r.U.d.

```
56      pessoa = banco.execute(
57        'SELECT * FROM pessoa WHERE id = (?)',
58        [id]
59      ).fetchone()
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']
66      banco = db.get_db()
67      banco.execute(
68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
```

# c.r.U.d.

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

# c.r.u.D.

```
 1 import os
 2 import db
 3
 4 from flask import Flask, render_template, reques
 5
 6
 7 def create_app():
 8   app = Flask(__name__)
 9
10   try:
11     os.makedirs(app.instance_path)
12   except OSError:
13     pass
14
15   app.config.from_mapping(
16     DATABASE=os.path.join(app.instance_path, f'{
17   )
```

# c.r.u.D.

```
65      nome = request.form['nome']
66      banco = db.get_db()
67      banco.execute(

68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
72      return redirect("listar")
73
74    @app.route('/<id>/deletar')
75    def deletar(id):
76      banco = db.get_db()
77      pessoa = banco.execute(
78        'DELETE FROM pessoa WHERE id = (?)',
79        [id]
80      )
81      banco.commit()
```

# c.r.u.D.

```
66      banco = db.get_db()
67      banco.execute(

68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
72      return redirect("listar")
73
74    @app.route('/<id>/deletar')
75    def deletar(id):
76      banco = db.get_db()
77      pessoa = banco.execute(
78        'DELETE FROM pessoa WHERE id = (?)',
79        [id]
80      )
81      banco.commit()
82      return redirect(url_for("listar"))
```

# c.r.u.D.

```
66      banco = db.get_db()
67      banco.execute(
68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
72      return redirect("listar")
73
74    @app.route('/<id>/deletar')
75    def deletar(id):
76      banco = db.get_db()
77      pessoa = banco.execute(
78        'DELETE FROM pessoa WHERE id = (?)',
79        [id]
80      )
81      banco.commit()
```

# c.r.u.D.

```
66      banco = db.get_db()
67      banco.execute(
68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
72      return redirect("listar")
73
74    @app.route('/<id>/deletar')
75    def deletar(id):
76      banco = db.get_db()
77      pessoa = banco.execute(
78        'DELETE FROM pessoa WHERE id = (?)',
79        [id]
80      )
81      banco.commit()
82      return redirect(url_for("listar"))
```

# c.r.u.D.

```
66      banco = db.get_db()
67      banco.execute(

68        'UPDATE pessoa SET nome = ? WHERE id = ?',
69        (nome, id)
70      )
71      banco.commit()
72      return redirect("listar")
73
74    @app.route('/<id>/deletar')
75    def deletar(id):
76      banco = db.get_db()
77      pessoa = banco.execute(
78        'DELETE FROM pessoa WHERE id = (?)',
79        [id]
80      )
81      banco.commit()
82      return redirect(url_for("listar"))
```

# c.r.u.D.

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

linkedin.com/in/jordyaraujo

jordyaraujo.github.io