

```
1 from logging import error
2
3 from flask import Flask, request
4 from flask.helpers import url_for
5 from flask.templating import render_template
6 from flask.wrappers import Request
7 from flask_migrate import Migrate
8 from werkzeug.utils import redirect
9
10 from database import db
11 from forms import PersonaForm
12 from models import Persona
13
14 app = Flask(__name__) #__name__ es el valor del nombre del archivo app.py
15
16 #Configuracion de la bd
17 USER_DB = 'postgres'
18 PASS_DB = 'root'
19 URL_DB = 'localhost'
20 NAME_DB = 'sap_flask_db'
21 FULL_URL_DB = f'postgresql://{USER_DB}:{PASS_DB}@{URL_DB}/{NAME_DB}'
22
23 #Configuracion del objeto
24 app.config['SQLALCHEMY_DATABASE_URI'] = FULL_URL_DB #Conecta la DB
25 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False # Evita modificaciones
    constantemente
26
27 #Inicializamos el objeto
28 db.init_app(app) #con esto se evita un problema circular
29
30 #Configurar flask-migrate
31 #Esto realiza las migraciones
32 migrate = Migrate()
33 migrate.init_app(app, db)
34
35 #Configurar flask-wtf
36 app.config [ 'SECRET_KEY' ] = 'llave_secreta'
37
38
39 @app.route('/')
40 @app.route('/index')
41 @app.route('/index.html')
42 def inicio():
43     #Listado de personas
44     personas = Persona.query.all()
45     total_personas = Persona.query.count()
46     #app.logger.debug(f'Listado Personas: {personas}')
47     #app.logger.debug(f'Listado Total Personas: {total_personas}')
48     app.logger.warning(f'Listado Personas: {personas}')
49     app.logger.warning(f'Listado Total Personas: {total_personas}')
50     return render_template('index.html', personas = personas, total_personas =
total_personas)
51
52 @app.route('/ver/<int:id>')
53 #Se especifica el tipo de dato y el nombre: <int:id> int: integer(por default es str)
    y id: nombre de variable
54 def ver_detalle(id):
55     #Recuperamos la persona segun el id proporcionado
56     persona = Persona.query.get_or_404(id)
57     app.logger.warning(f'Ver Persona: {persona}')
```

```
58     return render_template('detalle.html', persona=persona)
59
60 @app.route('/agregar', methods = ['GET', 'POST'])
61 def agregar_persona():
62     persona = Persona()
63     personaForm = PersonaForm(obj = persona) #obj especifica la clase de modelo a
        utilizar
64     if request.method == 'POST':
65         if personaForm.validate_on_submit:
66             personaForm.populate_obj(persona) #se carga el objeto persona con los
        datos del formulario
67             app.logger.warning(f'Persona Agregada: {persona}')
68             #Insertamos el nuevo REGISTRO
69             db.session.add(persona)
70             db.session.commit()
71             return redirect(url_for('inicio'))
72     return render_template('agregar.html', forma = personaForm)
73
74 @app.route('/editar/<int:id>', methods = ['GET', 'POST'])
75 def editar_persona(id):
76     persona = Persona.query.get_or_404(id)
77     personaForm = PersonaForm(obj = persona)
78
79     if request.method == 'POST':
80         if personaForm.validate_on_submit:
81             personaForm.populate_obj(persona)
82             app.logger.warning(f'Persona Editada: {persona}')
83             db.session.commit()
84             return redirect(url_for('inicio'))
85
86     return render_template('editar.html', forma = personaForm)
87
88 @app.route('/eliminar/<int:id>')
89 def eliminar(id):
90     persona = Persona.query.get_or_404(id)
91     db.session.delete(persona)
92     db.session.commit()
93
94     return redirect(url_for('inicio'))
```