**FLASK y BASE DE DATOS**

1. Crear carpeta y abrir en visual.

2. Crear archivo app.py.

3. Abrir terminal, usar código para crear carpeta venv y activar entorno virtual.

4. pip install flask y verificar la instalación con pip freeze.

5. Importar librería de Flask en app.py (from flask import Flask, render\_template, request).

6. app = Flask(\_\_name\_\_)

7. @app.route(“/”)

def index():

return render\_template(´index.html´)

8. if \_\_name\_\_ == ´\_\_main\_\_´:

app.run(debug=True)

9. Crear carpeta de templates y dentro el archivo index.html.

10. flask run (en terminal).

11. Instalar sqlai3 y sqlalchemy. (pip install sqlalchemy).

12. Llamar a la librería en app.py (from flask\_sqlalchemy import SQLAlchemy).

13. app.config[‘SQLALCHEMY\_DATABASE\_URI’] = ‘sqlite:///db.sqlite3’

app.config[‘SQLALCHEMY\_TRACK\_MODIFICATIONS’] = False

(poner debajo de app = Flask(\_\_name\_\_)).

14. Usar la app db.browser para crear y visualizar la base de datos.

15. Crear un objeto: db = SQLAlchemy(app). Pegar en terminal **pip install Flask-SQLAlchemy**

16. Crear los modelos de la base de datos:

class Emprendimientos(db.Model):

id = db.Column(db.Integrer, primary\_key=True)

nombre = db.Column(db.String(80), unique=True, nullable=False)

descripcion = db.Column(db.String(200), nullable=False)

contacto = db.Column(db.String(10), nullable=False)

lat = db.Column(db.Float(200), nullable=False)

lon = db.Column(db.Float(200), nullable=False)

def \_\_init\_\_(self, nombre, descripcion, contacto, lat, lon):

self.nombre = nombre

self.descripcion = descripcion

self.contacto = contacto

self.lat = lat

self.lon = lon

17. Agregar db.create\_all() en “if\_\_name\_\_...”

18. Crear un formulario, creando un nuevo archivo dentro de templates que se llame ‘registro.html’.

19. Usar form action method post. Tiene label e input. Llenar los espacios según lo definido en def\_\_init\_\_

20. Crear nueva ruta para registro.

@app.route(‘/registro’)

def registro():

return render\_template(´registro.html´)

21. Pegar el link de tailwind a nuestro html y hermosear el formulario.

22. En la ruta de registro agregar , methods = [‘GET’, ‘POST’]). En la def registro ():

If request.method == ‘POST’:

nombre = request.form[‘nombre’] y así con cada variable.

23. Pegar las variables del formulario en app.py en el def de registro para tener la info en el backend.

24. Crear el objeto emprendimiento:

emprendimiento = Emprendimientos(nombre, descripción, contacto, lat, lon)

db.session.add(emprendimiento)

db.session.commit()

return render\_template

25. @app.route(‘/emprendimientos’) (crear ruta para que aparezca el listado de registros).

def emprendimientos():

emprendimientos = Emprendimientos.query.all()

return render\_template(‘emprendimientos.html’, emprendimientos=emprendimientos)

26. Crear en templates emprendimientos.html. Para mostrar los registros:

{% for emprendimiento in emprendimientos %}

<h2>{{ emprendimientos.nombre }} </h2> 🡪 así con cada variable.

{%endfor%}