

## **Actividad de Aprendizaje 3 python**

### **Modelado y Visualización– Etapa I y II**

---

#### **Autores:**

Alejandro Restrepo Duque

Xiodanny Vásquez Holguin

Andrea Estefanía Pabón Garcés

**Ficha:** 2828526

**Fecha:** 21-03-2025

## **Informe de la Actividad de Aprendizaje 3 - Python**

### **Índice**

1. Introducción
2. Objetivos
  - Objetivo General
  - Objetivos Específicos
3. Desarrollo del Proyecto
  - Configuración y Desarrollo con Flask
  - Base de Datos
  - Serialización de Datos
4. Resultados y Conclusiones
5. Evidencias

## 1. Introducción

Python es un lenguaje ampliamente utilizado en el desarrollo de aplicaciones web debido a su simplicidad y versatilidad. En esta actividad, se ha trabajado con Flask, un micro-framework que permite desarrollar aplicaciones web de manera ágil. El proyecto desarrollado tiene como objetivo la captura, almacenamiento y análisis de datos de empleados en una base de datos, utilizando SQLite como sistema de gestión de bases de datos relacional, gestionado mediante SQLAlchemy.

## 2. Objetivos

### 2.1 Objetivo General

Desarrollar una aplicación web utilizando Flask para gestionar formularios de evaluación de empleados y almacenar los datos en una base de datos.

### 2.2 Objetivos Específicos

- Implementar una conexión a una base de datos SQLite para almacenar los datos capturados.
- Aplicar conceptos de serialización de datos en formatos CSV y JSON.
- Analizar los datos utilizando librerías de Python como Pandas y NumPy.

## 3. Desarrollo del Proyecto

### 3.1 Configuración y Desarrollo con Flask

Se ha implementado una aplicación web con Flask que incluye varios formularios para la captura de datos en diferentes aspectos de la "Rueda de la Vida":

- **Empleado:** Datos personales y laborales.
- **Amor:** Aspectos emocionales y relaciones interpersonales.
- **Economía:** Administración financiera y hábitos de ahorro.
- **Salud:** Estado físico y bienestar mental.
- **Desarrollo Personal:** Aprendizaje y crecimiento profesional.

```

3 from datetime import datetime
4 import pandas as pd
5 import sqlite3
6
7 app = Flask(__name__)
8
9 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///respuestas.db'
10 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
11 db = SQLAlchemy(app)
12
13 # Modelo completo para las respuestas
14 You, hace 15 minutos | 2 authors (You and one other)
15 class Respuesta(db.Model):
16     id = db.Column(db.Integer, primary_key=True)
17     formulario = db.Column(db.String(50), nullable=False)
18
19     # Datos del empleado
20     nombre = db.Column(db.String(50), nullable=True)
21     apellido = db.Column(db.String(50), nullable=True)
22     vivienda = db.Column(db.String(100), nullable=True)
23     departamento = db.Column(db.String(50), nullable=True)
24     estado_civil = db.Column(db.String(20), nullable=True)
25     genero = db.Column(db.String(20), nullable=True)
26     edad = db.Column(db.Integer, nullable=True)
27     salario = db.Column(db.Integer, nullable=True)
28     cargo = db.Column(db.String(50), nullable=True)
29     estrato = db.Column(db.Integer, nullable=True)
30
31     # Sección Amor
32     relacion = db.Column(db.Integer, nullable=True)

```

```

44
45 # Ruta para visualizar respuestas
46 @app.route("/ver_respuestas")
47 def ver_respuestas():
48     respuestas = Respuesta.query.all()
49     return render_template("ver_respuestas.html", respuestas=respuestas)
50
51 # Ruta para eliminar respuesta
52 @app.route("/eliminar_respuesta/<int:id>", methods=["POST"])
53 def eliminar_respuesta(id):
54     respuesta = Respuesta.query.get_or_404(id)
55     db.session.delete(respuesta)
56     db.session.commit()
57     return redirect(url_for("ver_respuestas"))
58

```

### 3.2 Base de Datos

La base de datos utilizada en este proyecto es **SQLite**, un sistema de gestión de bases de datos ligero y eficiente para aplicaciones locales. Se ha creado la base de datos respuestas.db dentro de la carpeta instance del proyecto. La estructura de la tabla principal se define de la siguiente manera:

```
respuestas.db M X
formulario-ruedadelavida > instance > respuestas.db
You, hace 8 horas | 2 authors (alejgto and one other)
1 SQLite format 3
2
3
4
5 id INTEGER NOT NULL,
6 formulario VARCHAR(50) NOT NULL,
7 nombre VARCHAR(50),
8 apellido VARCHAR(50),
9 vivienda VARCHAR(100),
10 departamento VARCHAR(50),
11 estado_civil VARCHAR(20),
12 genero VARCHAR(20),
13 edad INTEGER,
14 salario INTEGER,
15 cargo VARCHAR(50),
16 estrato INTEGER,
17 relacion INTEGER,
18 expresion_amor INTEGER,
19 apertura_relaciones INTEGER,
20 esfuerzo_relaciones INTEGER,
21 claridad_amor INTEGER,
22 fecha_envio DATETIME,
23 PRIMARY KEY (id)
24
25
26
27 2025-03-20 19:29:29.457676Z
28
```

Los datos ingresados por los usuarios a través del formulario web se almacenan en esta tabla, permitiendo su posterior análisis y visualización.

### 3.3 Serialización de Datos

Se han implementado procesos para la conversión de los datos almacenados en la base de datos a formatos **CSV y JSON**, facilitando su análisis posterior. Para ello, se utilizan las librerías Pandas y JSON de Python.

```
160 # EXPORTAR DATOS CSV y JSON
161 # =====
162
163 @app.route("/exportar_csv")
164 def exportar_csv():
165     try:
166         with app.app_context():
167             df = pd.read_sql("SELECT * FROM Respuesta", db.engine)
168             csv_data = df.to_csv(index=False)
169
170             response = Response(csv_data, content_type="text/csv")
171             response.headers["Content-Disposition"] = "attachment; filename=respuestas.csv"
172             return response
173     except Exception as e:
174         return f"Error al exportar CSV: {e}"
175
176 @app.route("/exportar_json")
177 def exportar_json():
178     try:
179         with app.app_context():
180             df = pd.read_sql("SELECT * FROM Respuesta", db.engine)
181             json_data = df.to_json(orient="records", indent=4)
182
183             response = Response(json_data, content_type="application/json")
184             response.headers["Content-Disposition"] = "attachment; filename=respuestas.json"
185             return response
186     except Exception as e:
187         return f"Error al exportar JSON: {e}"
188
```

## 4. Resultados y Conclusiones

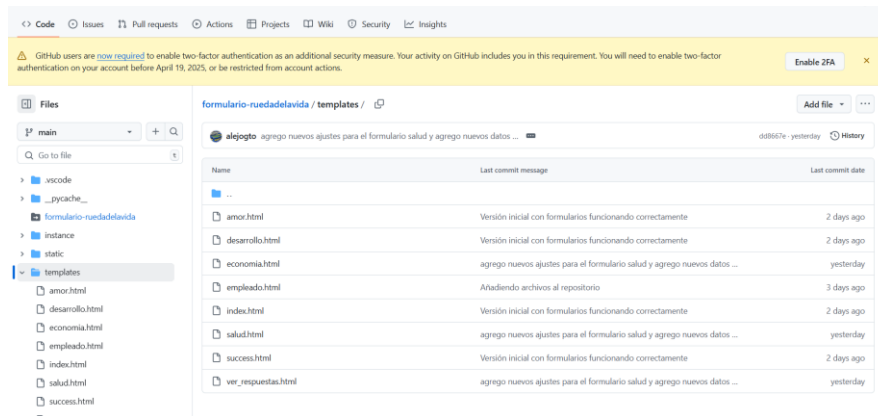
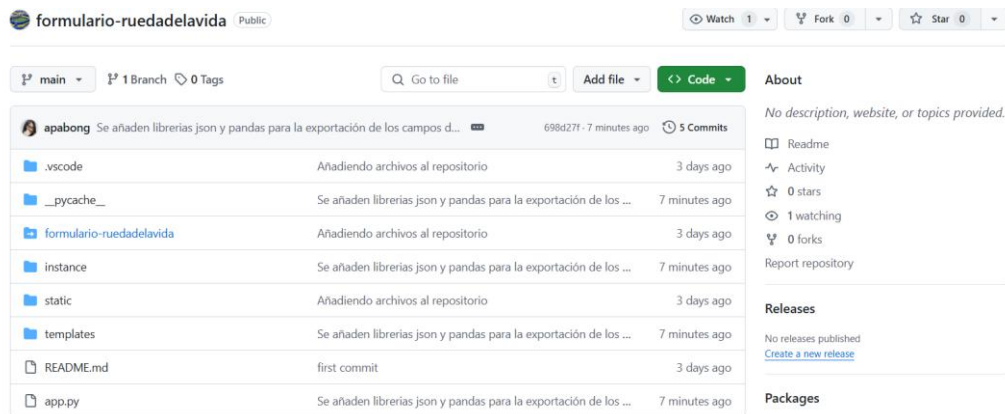
El desarrollo de esta actividad ha permitido:

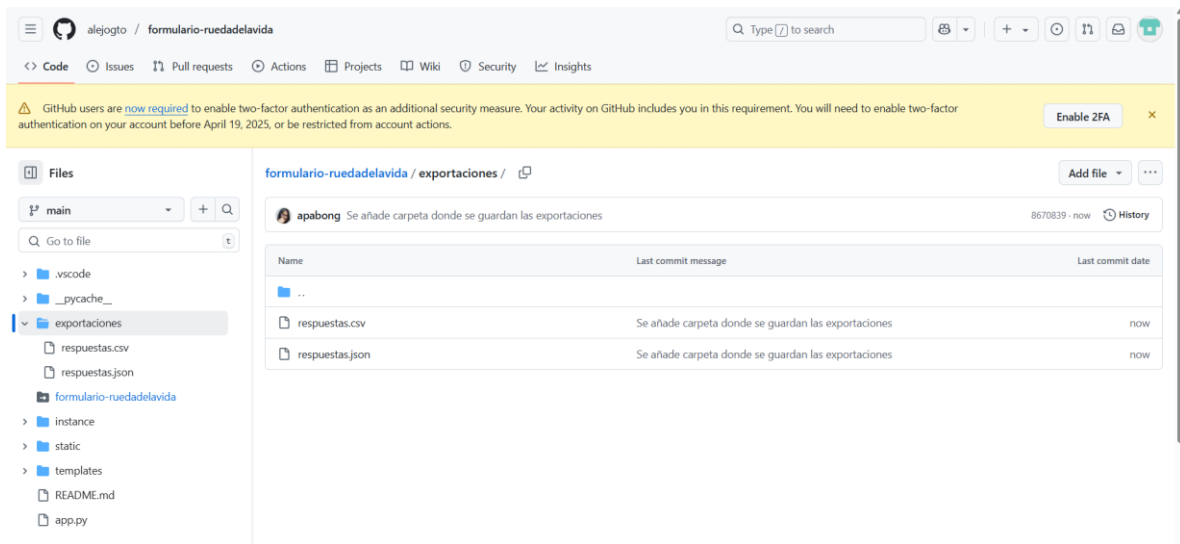
- Implementar una aplicación web funcional con Flask.
- Conectar y manipular una base de datos SQLite desde Python.
- Aplicar técnicas de serialización de datos.

Este proyecto proporciona una herramienta útil para evaluar el bienestar de los empleados y mejorar la toma de decisiones en la gestión de recursos humanos.

## 5. Evidencias

- Código fuente del proyecto en un repositorio de GitHub.





- Capturas de pantalla de la aplicación en funcionamiento.

127.0.0.1:5000

### Formulario de Empleado

Nombre

Apellido

Vivienda

Departamento

Estado Civil

Género

Edad

Salario

Cargo

Estrato Socioeconómico

Enviar

127.0.0.1:5000/ver\_respuestas

### Respuestas Guardadas

ID	Formulario	Nombre	Apellido	Edad	Salario	Estado Civil	Género	Cargo	Estrato	Fecha	Acciones
1	empleado	Andrea	Garcés	22	8,000,000.00	soltero	femenino	Asistente Soporte	2	2025-03-21	Eliminar
2	empleado	Xiodanny	Vasquez	29	10,000,000.00	soltero	femenino	Programador	2	2025-03-21	Eliminar
3	empleado	Alejandro	Restrepo	29	80,000,000.00	soltero	masculino	Programador	3	2025-03-21	Eliminar

Volver al inicio Descargar CSV Descargar JSON Actualizar Tabla

- Archivos CSV y JSON generados.

Este equipo > Windows (C:) > FormularioRuedadelavida > formulario-ruedadelavida > exportaciones				
Ordenar Ver				
Nombre	Fecha de modificación	Tipo	Tamaño	
respuestas.csv	21/03/2025 7:30 a. m.	Archivo de valores...	1 KB	
respuestas.json	21/03/2025 7:31 a. m.	Archivo JSON	4 KB	