

Actividad de Aprendizaje 4-WebSite con BD

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 Desarrollo Web con Flask y Base de Datos

1. Introducción

En este documento se describe el desarrollo de una aplicación web con Flask y una base de datos SQLite. La aplicación permite capturar, almacenar y exportar datos de usuarios mediante un formulario web.

127.0.0.1:5000

Formulario de Empleado

Nombre

Apellido

Vivienda

Departamento

Estado Civil

Soltero

Género

Masculino

Edad

Salario

Cargo

Estrato Socioeconómico

Enviar

2. Objetivos

- Implementar una aplicación web con Flask.
- Utilizar SQLite para el almacenamiento de datos.
- Exportar datos en formato CSV y JSON.
- Gestionar el código con GitHub.

3. Desarrollo

3.1 Estructura y Diseño del Sitio Web

El proyecto sigue la estructura estándar de Flask, con carpetas para plantillas HTML (templates), archivos estáticos (static) y la aplicación principal (app.py).

```
alejogto, anteayer | 2 authors (Xiodanny-Vasquez and one other)
1 <!DOCTYPE html> Xiodanny-Vasquez, hace 3 días • Añadiendo archivos al repositorio
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Selecciona un Formulario</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8   <link rel="stylesheet" href="{{ url_for('static', filename='estilos.css') }}">
9 </head>
10 <body>
11   <div class="container1">
12     <h1>Selecciona un Formulario</h1>
13     <br>
14     <ul>
15       <li><a href="{{ url_for('empleado') }}">Formulario de Empleado</a></li>
16       <li><a href="{{ url_for('amor') }}">Formulario de Amor</a></li>
17       <li><a href="{{ url_for('economia') }}">Formulario de Economía</a></li>
18       <li><a href="{{ url_for('salud') }}">Formulario de Salud</a></li>
19       <li><a href="{{ url_for('desarrollo') }}">Formulario de Desarrollo Personal</a></li>
20       <div class="text-center mt-4">
21         <a href="{{ url_for('ver_respuestas') }}" class="btn btn-info">Ver Respuestas Guardadas</a>
22       </div>
23     </ul>
24   </div>
25 </body>
26 </html>
```

3.2 Captura de Datos con Flask

Los datos se capturan mediante un formulario HTML y se envían al servidor a través de una petición POST.

```
@app.route("/", methods=["GET", "POST"])
def empleado():
    if request.method == "POST":
        nueva_respuesta = Respuesta(
            formulario="empleado",
            nombre=request.form["nombre"],
            apellido=request.form["apellido"],
            vivienda=request.form["vivienda"],
            departamento=request.form["departamento"],
            estado_civil=request.form["estado_civil"],
            genero=request.form["genero"],
            edad=request.form["edad"],
            salario=request.form["salario"],
            cargo=request.form["cargo"],
            estrato=request.form["estrato"]
        )
        db.session.add(nueva_respuesta)
        db.session.commit()
        return redirect(url_for("amor", id=nueva_respuesta.id))

    return render_template("empleado.html")
```

3.3 Integración Frontend-Backend

El formulario HTML se conecta con Flask para enviar los datos ingresados por el usuario.

```
@app.route("/desarrollo/<int:id>", methods=["GET", "POST"])
def desarrollo(id):
    respuesta = Respuesta.query.get_or_404(id)
    if request.method == "POST":
        respuesta.aprendizaje_desarrollo = request.form["aprendizaje_desarrollo"]
        respuesta.claridad_metas_suenos = request.form["claridad_metas_suenos"]
        respuesta.esfuerzo_maximo_potencial = request.form["esfuerzo_maximo_potencial"]
        respuesta.pasion_motivacion = request.form["pasion_motivacion"]
        respuesta.crecimiento_personal = request.form["crecimiento_personal"]
        db.session.commit()
        return redirect(url_for("success"))

    return render_template("desarrollo.html", id=id)

@app.route("/success")
def success():
    return render_template("success.html")

@app.route("/ver_respuestas")
def ver_respuestas():
    respuestas = Respuesta.query.all()
    return render_template("ver_respuestas.html", respuestas=respuestas)

@app.route("/eliminar_respuesta/<int:id>", methods=["POST"])
def eliminar_respuesta(id):
    respuesta = Respuesta.query.get_or_404(id)
    db.session.delete(respuesta)
    db.session.commit()
    return redirect(url_for("ver_respuestas"))
```

3.4 Serialización y Almacenamiento de Datos

Los datos capturados se almacenan en una base de datos SQLite. Además, se implementó la exportación de datos en formatos CSV y JSON utilizando la biblioteca panda.

- Código donde se almacena la información en SQLite

```
from flask import Flask, render_template, request, redirect, url_for, Response
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
import pandas as pd
import os # Importa os para manejar directorios

app = Flask(__name__)

app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///respuestas.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

# Modelo completo para las respuestas
class Respuesta(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    formulario = db.Column(db.String(50), nullable=False)

    # Datos del empleado
    nombre = db.Column(db.String(50), nullable=True)
    apellido = db.Column(db.String(50), nullable=True)
    vivienda = db.Column(db.String(100), nullable=True)
    departamento = db.Column(db.String(50), nullable=True)
    estado_civil = db.Column(db.String(20), nullable=True)
    genero = db.Column(db.String(20), nullable=True)
    edad = db.Column(db.Integer, nullable=True)
    salario = db.Column(db.Integer, nullable=True)
    cargo = db.Column(db.String(50), nullable=True)
    estrato = db.Column(db.Integer, nullable=True)
```

- Código de la función que exporta datos a CSV/JSON.

```
@app.route("/exportar_csv")
def exportar_csv():
    try:
        with app.app_context():
            df = pd.read_sql("SELECT * FROM Respuesta", db.engine)

            carpeta = "exportaciones"
            if not os.path.exists(carpeta):
                os.makedirs(carpeta)

            ruta_archivo = os.path.join(carpeta, "respuestas.csv")
            df.to_csv(ruta_archivo, index=False)

            return f"Archivo CSV exportado correctamente en: {ruta_archivo}"
    except Exception as e:
        return f"Error al exportar CSV: {e}"

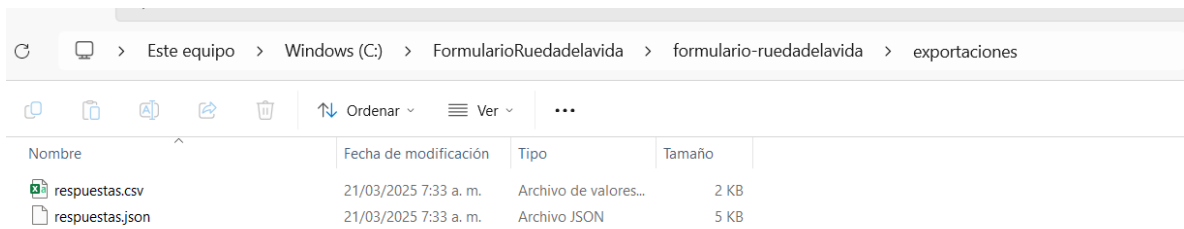
@app.route("/exportar_json")
def exportar_json():
    try:
        with app.app_context():
            df = pd.read_sql("SELECT * FROM Respuesta", db.engine)

            carpeta = "exportaciones"
            if not os.path.exists(carpeta):
                os.makedirs(carpeta)

            ruta_archivo = os.path.join(carpeta, "respuestas.json")
            df.to_json(ruta_archivo, orient="records", indent=4)

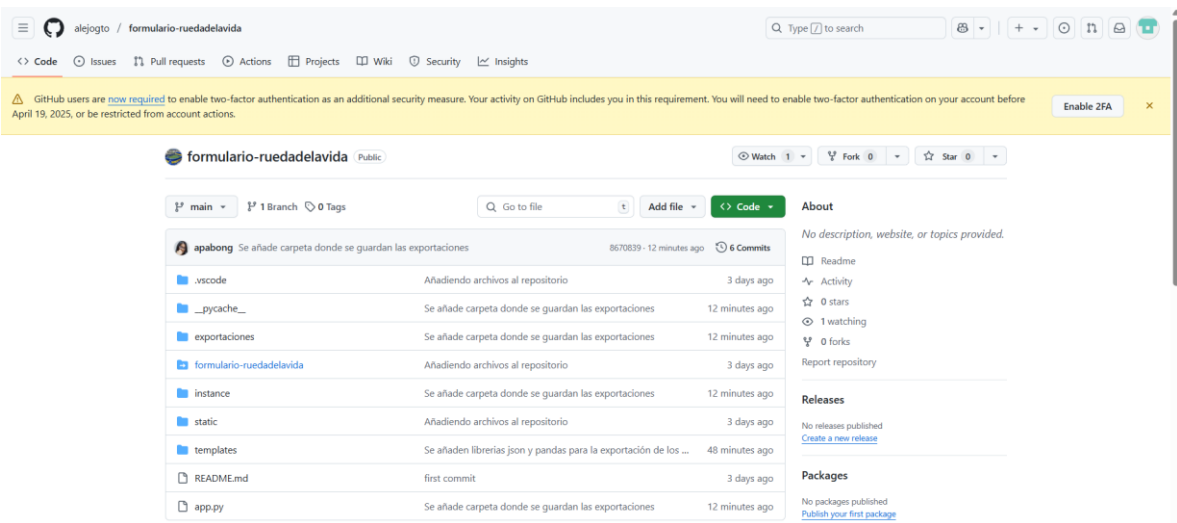
            return f"Archivo JSON exportado correctamente en: {ruta_archivo}"
    except Exception as e:
        return f"Error al exportar JSON: {e}"
```

- Captura de los archivos generados en la carpeta exportaciones/.



3.5 Gestión del Código en GitHub

El código del proyecto se encuentra en un repositorio de GitHub, donde se pueden visualizar los archivos y el historial de cambios.



- Historial de commits mostrando la evolución del proyecto.



4. Resultados

El desarrollo del proyecto resultó en una aplicación web funcional que permite capturar datos mediante un formulario, almacenarlos en una base de datos y exportarlos en diferentes formatos.

127.0.0.1:5000/amor/5

Formulario sobre Amor y Relaciones

¿Cómo califica sus relaciones personales?

¿Expresa amor claramente?

¿Está abierto a nuevas relaciones?

¿Se esfuerza en mantener sus relaciones?

¿Tiene claridad sobre lo que busca en el amor?

Siguiente

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
4	empleado	David	Espinel	23	9,000,000,000.00	soltero	masculino	Auxiliar Logistico	2	2025-03-21	Eliminar
5	empleado	Maribel	Garcia	22	800,000,000.00	soltero	femenino	Auxiliar de Bodega	3	2025-03-21	Eliminar

Volver al inicio

Descargar CSV

Descargar JSON

Actualizar Tabla

5. Evidencias

Se incluyen capturas de pantalla del funcionamiento de la aplicación, la base de datos y los archivos exportados.

The screenshot displays two parts of the application. The top part is a web browser window showing a table titled "Respuestas Guardadas". The table has 12 columns: ID, Formulario, Nombre, Apellido, Edad, Salario, Estado Civil, Género, Cargo, Estrato, Fecha, and Acciones. It contains 5 rows of employee data. Below the table are four buttons: "Volver al inicio", "Descargar CSV", "Descargar JSON", and "Actualizar Tabla". The bottom part of the screenshot shows a Windows File Explorer window with the path "Este equipo > Windows (C:) > FormularioRuedadelavida > formulario-ruedadelavida > exportaciones". It displays two files: "respuestas.csv" (2 KB) and "respuestas.json" (5 KB), both modified on 21/03/2025 at 7:33 a.m.

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
4	empleado	David	Espinel	23	9,000,000,000.00	soltero	masculino	Auxiliar Logistico	2	2025-03-21	Eliminar
5	empleado	Maribel	Garcia	22	800,000,000.00	soltero	femenino	Auxiliar de Bodega	3	2025-03-21	Eliminar

Buttons: Volver al inicio, Descargar CSV, Descargar JSON, Actualizar Tabla

File Explorer Path: Este equipo > Windows (C:) > FormularioRuedadelavida > formulario-ruedadelavida > exportaciones

Nombre	Fecha de modificación	Tipo	Tamaño
respuestas.csv	21/03/2025 7:33 a. m.	Archivo de valores...	2 KB
respuestas.json	21/03/2025 7:33 a. m.	Archivo JSON	5 KB

6. Conclusiones

El proyecto permitió la integración de Flask con bases de datos y la gestión del código con GitHub, demostrando la importancia del backend en aplicaciones web.

7. Referencias

- Documentación oficial de Flask: <https://flask.palletsprojects.com/>
- Documentación de SQLite: <https://www.sqlite.org/docs.html>
- Uso de pandas para exportación de datos: <https://pandas.pydata.org/>