**Manual de Usuario y Desarrollador**

**Proyecto: Análisis de Regresión de Poisson**

**AUTOR:** Juan Carlos Anquise Vargas

**Introducción:**

Este proyecto implementa una aplicación web desarrollada con Flask que permite realizar un análisis de regresión de Poisson utilizando datos generados automáticamente. La aplicación también permite descargar los datos generados en formato CSV o Excel.

**Estructura del Proyecto**

El proyecto está organizado en la siguiente estructura de carpetas:

project/

│

├── static/ # Recursos estáticos como imágenes

│ └── img/

│ └── UNA-Puno-2.png # Imagen de fondo

│

├── templates/ # Plantillas HTML para la aplicación Flask

│ └── index.html # Página principal

│

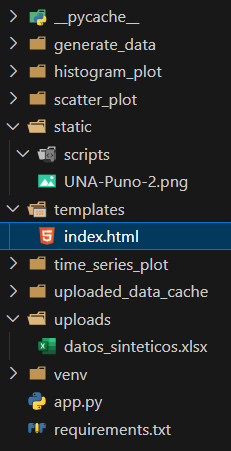
├── data/ # Carpeta para almacenar datos generados o subidos

│

├── app.py # Archivo principal que ejecuta el servidor Flask

├── requirements.txt # Dependencias del proyecto

└── README.md # Instrucciones y descripción del proyecto



**Instalación:**

**Requisitos previos**

1. Python 3.8 o superior.
2. Gestor de paquetes pip.
3. Entorno virtual (opcional) para aislar las dependencias del proyecto.

**Pasos de instalación**

1. Clona el repositorio del proyecto o descárgalo como archivo ZIP.
2. Navega a la carpeta del proyecto desde la terminal:

* bash
* cd project

1. Crea y activa un entorno virtual (opcional):
   * bash
   * python -m venv venv
   * En Windows: venv\Scripts\activate
2. Instala las dependencias del proyecto:

* bash
* pip install -r requirements.txt

1. Ejecuta la aplicación Flask:

* bash
* python app.py

1. Accede a la aplicación en tu navegador web utilizando la URL:

* http://127.0.0.1:5000/

**Uso de la Aplicación:**

**Inicio:**

1. Al abrir la aplicación en tu navegador, verás una interfaz gráfica con los siguientes elementos:

* Encabezados que identifican a la Universidad Nacional del Altiplano.
* Un fondo personalizado con la imagen proporcionada.



1. Puedes interactuar con el formulario para generar datos o descargar datos existentes.

Generación de Datos

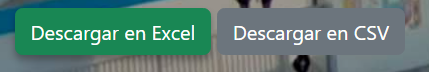
* Completa el formulario para indicar el número de muestras que deseas generar.
* Haz clic en el botón de generación.
* La aplicación generará un archivo `generated\_data.csv` en la carpeta `data/` con las columnas predeterminadas.





Descarga de Datos

* Haz clic en el botón de descarga.
* Selecciona el formato deseado: CSV o Excel.
* El archivo se descargará a tu computadora.

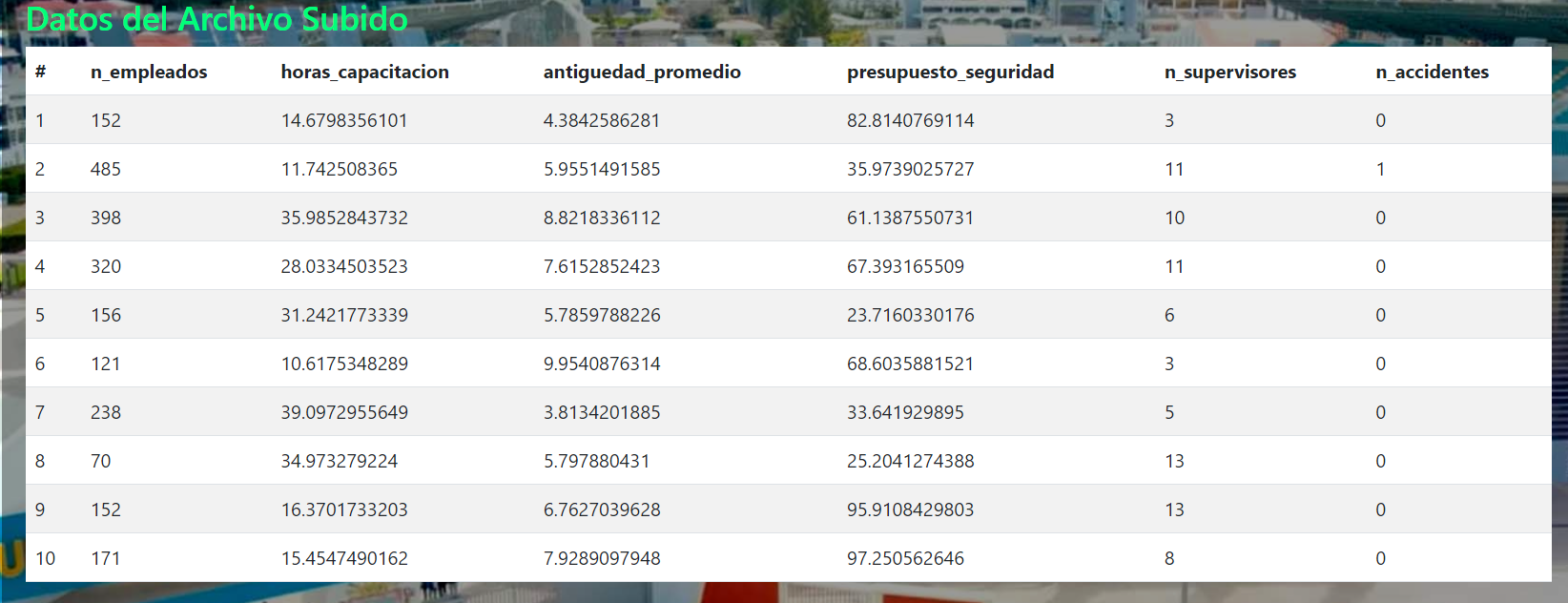


Cargar Archivos:

* Al cargar el archivo se visualiza el peso del archivo



* Analizamos y visualizamos los datos del archivo



* Visualizamos los resultados del archivo



* Seleccionamos variables para generar los graficos



**Archivos Clave**

**1. app.py**

El archivo principal que ejecuta el servidor Flask. Contiene las rutas para:

- Cargar la página principal (`/`).

- Generar datos en formato CSV (`/generate\_data`).

- Descargar datos en diferentes formatos (`/download\_data`).

**Código destacado:**

python

@app.route('/generate\_data', methods=['POST'])

def generate\_data():

n\_samples = int(request.form.get('n\_samples', 100))

data = pd.DataFrame({

'n\_empleados': [5] \* n\_samples,

'horas\_capacitacion': [10] \* n\_samples,

'antiguedad\_promedio': [2.5] \* n\_samples,

'presupuesto\_seguridad': [1000] \* n\_samples,

'n\_supervisores': [3] \* n\_samples,

'n\_accidentes': [1] \* n\_samples

})

data.to\_csv('data/generated\_data.csv', index=False)

return data.to\_json(orient='records')

```

**2. index.html**

La plantilla HTML principal. Contiene la estructura de la interfaz del usuario y referencias a archivos CSS y JavaScript.

Cambiar el fondo

1. Sustituye la imagen `UNA-Puno-2.png` en `static/img/` por la imagen deseada.

2. Asegúrate de que el archivo tenga el mismo nombre o actualiza la referencia en `styles.css`:

css

background-image: url('/static/img/nueva-imagen.png')

**Agregar nuevas columnas a los datos generados**

1. Edita la función `generate\_data` en `app.py`.

2. Agrega nuevas columnas al diccionario `data`.

**Mantenimiento**

**Solución de Problemas**

1. Error: “ModuleNotFoundError” al ejecutar `app.py`

- Verifica que las dependencias estén instaladas:

bash

pip install -r requirements.txt

2. Los estilos o scripts no se cargan

- Asegúrate de que la carpeta `static/` esté en la ubicación correcta y que los archivos tengan permisos adecuados.

3. El archivo generado no se descarga

- Verifica que la carpeta `data/` exista y tenga permisos de escritura.

**Extensiones Futuras**

1. Integración de modelos estadísticos: Implementar el análisis de regresión directamente en la aplicación.

2. Autenticación: Agregar una capa de autenticación para proteger el acceso.

3. Visualizaciones: Generar gráficas y reportes automáticos a partir de los datos generados.

**CODIGO FUENTE EN GITHUB**

Por el momento solamente analiza los datos sintéticos generados, una vez descargados y subir ese mismo archivo para visualizar los resultados del análisis

<https://github.com/JuanCarlosAnquise1007/codigo>