# EJERCICIO #5

Electiva IV

Jaime Andrés Martínez Ordoñez

Uniautónoma
DEL CAUCA

2025

**Implementación del IA en la web**

Cristhian Cañar
04/04/2025

# Taller #5 Implementación del IA en la web:

## 1. Explicación Código paso a paso:

### 1.1. Importación de librerías

El programa inicia cargando las librerías esenciales para su funcionamiento. Se utiliza **streamlit** para crear una interfaz web interactiva, pandas para gestionar estructuras de datos, **matplotlib.pyplot** para generar gráficos y **numpy** para manejar operaciones numéricas y generar datos aleatorios. Estas herramientas, como se muestra en la Figura 1, son fundamentales para el procesamiento de los datos y su visualización.

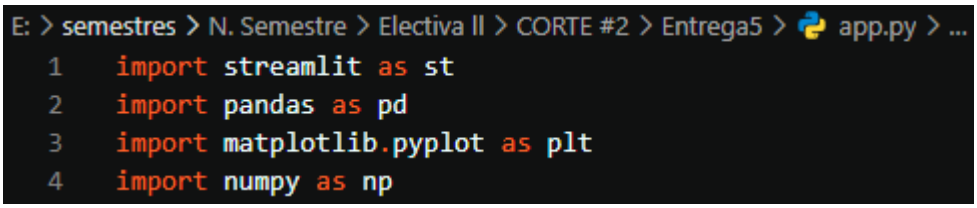


Fig 1. Librerías.

### 1.2. Título de la aplicación

A continuación, se define un encabezado principal con la función **st.title**, el cual se presenta en la parte superior de la aplicación web. Esto proporciona contexto inmediato al usuario sobre la funcionalidad de la interfaz. En la Figura 2, se aprecia cómo este título orienta visualmente al usuario desde el inicio.

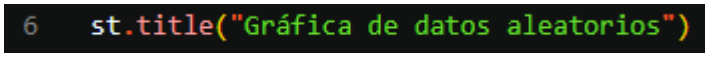


Fig 2. Título.

### 1.3. Slider para seleccionar la cantidad de datos

Se incorpora un **slider** que permite al usuario seleccionar dinámicamente cuántos datos desea generar, dentro de un rango de 5 a 50 valores, con un valor predeterminado de 10. Esta función interactiva, que puede observarse en la Figura 3, hace que la experiencia sea más flexible y personalizada según la necesidad del usuario.

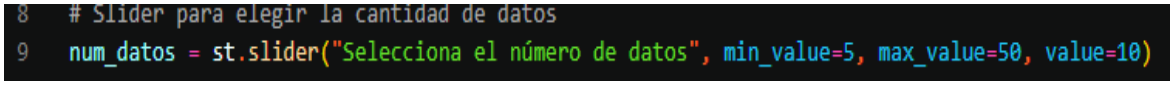


Fig 3. Cantidad de Datos.

## 1.4. Generar de datos aleatorios

Luego se generan dos arreglos: uno de tiempos y otro de voltajes. El arreglo de tiempos (tiempo) se crea con valores aleatorios ordenados entre 0 y 10 segundos, mientras que el de voltajes (voltaje) contiene valores entre 0 y 5V. Este proceso, evidenciado en la Figura 4, simula la adquisición de datos experimentales de forma sencilla.

```
11  # Generar datos aleatorios
12  tiempo = np.sort(np.random.uniform(0, 10, num_datos))  # Tiempos entre 0 y 10 segundos
13  voltaje = np.random.uniform(0, 5, num_datos)  # Voltajes entre 0 y 5V (puedes ajustar el rango)
```

Fig 4. Generar Datos Aleatorios.

## 1.5. Creación del DataFrame

Con los datos ya generados, se construye un **DataFrame** usando pandas, lo cual permite almacenar los tiempos y voltajes en una estructura tabular de dos columnas. Esta tabla, vista en la Figura 5, facilita el análisis y manipulación de los datos antes de ser graficados.

```
15  # Crear DataFrame
16  df = pd.DataFrame({
17      "Tiempo (s)": tiempo,
18      "Voltaje (V)": voltaje
19  })
```

Fig 5. DataFrame.

## 1.6. Visualización de los datos en forma de tabla

Posteriormente, el contenido del DataFrame se despliega en pantalla junto con un texto descriptivo. Esto permite que el usuario verifique la información generada de forma clara, tal como se aprecia en la Figura 6, antes de pasar a la etapa de visualización gráfica.

```
21  # Mostrar datos en una tabla
22  st.write("Datos generados aleatoriamente:")
23  st.dataframe(df)
24
```

Fig 6. Datos de la Tabla.

### 1.7. Creación y visualización de la gráfica

Finalmente, se crea una gráfica de línea utilizando **matplotlib**, donde se representa el voltaje en función del tiempo. Los puntos se marcan con círculos y se le asignan etiquetas a los ejes, así como un título al gráfico. Esta representación visual, ilustrada en la Figura 7, permite interpretar rápidamente cómo varía el voltaje respecto al tiempo.

```
25  # Graficar los datos
26  fig, ax = plt.subplots()
27  ax.plot(df["Tiempo (s)"], df["Voltaje (V)"], marker="o", linestyle="-")
28  ax.set_xlabel("Tiempo (s)")
29  ax.set_ylabel("Voltaje (V)")
30  ax.set_title("Voltaje vs. Tiempo")
31  st.pyplot(fig)
```

Fig 7. Visualización de la Gráfica.

## 2. Ejecución del comando

Después de ejecutar el comando **streamlit run app.py** en la terminal, se muestra un mensaje confirmando que la aplicación de **Streamlit** se ha iniciado correctamente, como se observa en la Figura 8. En este mensaje, se indican las direcciones URL generadas, permitiendo el acceso a la interfaz tanto desde el equipo local como desde otros dispositivos conectados a la misma red. Además, la terminal refleja el estado de ejecución del servidor, asegurando que la aplicación está lista para ser utilizada.

```
PROBLEMAS   SALIDA   CONSOLA DE DEPURACIÓN   TERMINAL   PUERTOS   MONITOR SERIE

PS E:\semestres\N. Semestre\Electiva ll\CORTE #2\proyect_web\streamlitApp> streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.6:8501
```

Fig 8. Ejecución de comando en el Terminal.

# 3. Interfaz de la aplicación y visualización de datos

## 3.1. Control deslizante para seleccionar el número de datos

En la parte superior de la interfaz se encuentra un control deslizante que permite al usuario seleccionar la cantidad de datos a generar, como se observa en la Figura 9. Este control define cuántos valores aleatorios de tiempo y voltaje se crearán, afectando tanto la tabla como la gráfica.

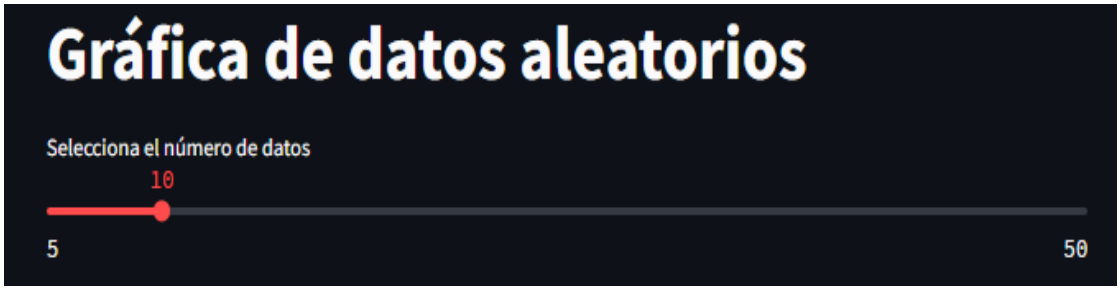


Fig 9. Seleccionador número de Datos

## 3.2. Tabla con los datos generados

Debajo del control deslizante, la aplicación muestra una tabla con los datos generados aleatoriamente. Esta tabla presenta dos columnas: "Tiempo (s)" y "Voltaje (V)", donde los valores de tiempo están ordenados de forma ascendente y los voltajes toman valores aleatorios entre 0 y 5V. La visualización de estos datos permite analizar la distribución de los valores antes de graficarlos.

Datos generados aleatoriamente:

| | Tiempo (s) | Voltaje (V) |
|---|---|---|
| 0 | 0.3302 | 3.1431 |
| 1 | 1.021 | 0.7276 |
| 2 | 1.3162 | 3.8008 |
| 3 | 3.124 | 4.9669 |
| 4 | 4.5033 | 0.2072 |
| 5 | 4.6623 | 3.6022 |
| 6 | 5.0773 | 3.3298 |
| 7 | 7.6627 | 4.4073 |
| 8 | 9.0857 | 0.3068 |
| 9 | 9.4918 | 1.5209 |

Fig 10. Tabla de Datos generados.

### 3.3. Gráfica de voltaje vs. Tiempo

En la parte inferior de la interfaz se encuentra la gráfica que representa los datos generados. Utilizando Matplotlib, la aplicación dibuja un gráfico de líneas donde el eje X representa el tiempo y el eje Y el voltaje. Se marcan puntos en cada dato para facilitar la visualización de la variación del voltaje a lo largo del tiempo. Esta gráfica permite observar tendencias y fluctuaciones en los valores obtenidos.

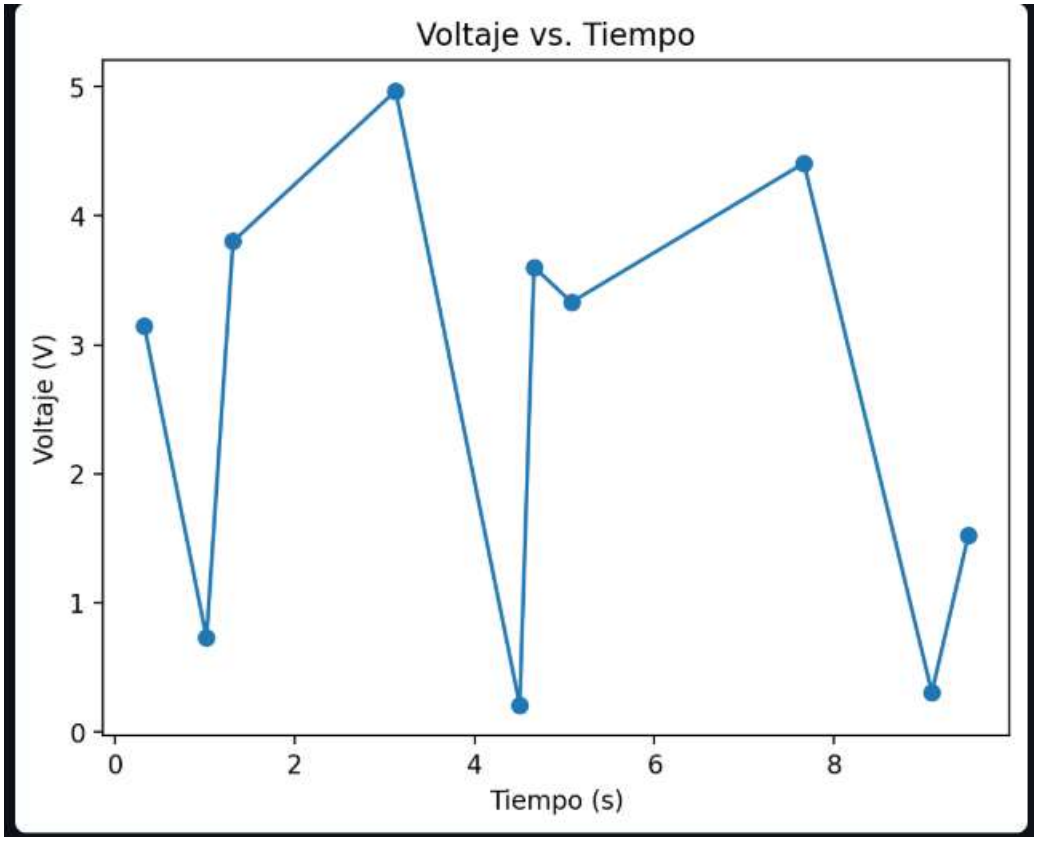


Fig 11. Gráfico de  Datos generados.

## 4. Enlace del repositorio público en GitHub.

- http://localhost:8501/
- https://github.com/andresmartinez444/taller-5.git

## 5. Bibliografía

### Referencias

https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html. (s.f.).
https://streamlit.io/gallery. (s.f.).