

## **Visualizador de Voltaje con streamlit**

Alex Javier Ramirez Pérez

Corporación Universitaria Autónoma del Cauca  
Ingeniería Electrónica  
Electiva II  
Cristhian Alejandro Cañar Muñoz  
6 de abril de 2025

## INTRODUCCION

Este informe presenta una aplicación web interactiva para la visualización de datos, la cual fue desarrollada utilizando Streamlit, el cual es un framework de Python de código abierto que permite transformar scripts de datos en aplicaciones web compatibles de forma rápida y sencilla.

La aplicación que desarrolle y que se detalla en este documento, tiene como objetivo graficar mediciones de Voltaje en función del Tiempo. Para ello, se combinó la capacidad de Streamlit para crear interfaces de usuario interactivas (como sliders para ajustar parámetros) con librerías como Pandas para la gestión de los datos, Matplotlib para generar las visualizaciones gráficas, Random para los valores aleatorios de voltaje y Numpy para crear los intervalos de tiempo. El resultado es una herramienta accesible que permite explorar cómo se comporta el voltaje a lo largo de un intervalo de tiempo definido.

## DESARROLLO (CODIGO)

```
# Importar librerías necesarias
import random as rd
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import streamlit as st
```

- Se importan las librerías necesarias para el desarrollo de la Web.
- **Random** para elegir los valores aleatorios de los voltajes.
- **Numpy** para crear el intervalo de tiempo en relación con los voltajes
- **Pandas** para la gestión de datos
- **Matplotlib.pyplot** para las visualizaciones de las graficas
- **Streamlit** para la interfaz web

```
st.title("Gráfica de datos con Streamlit")

# Crear lista para almacenar los datos de voltaje y tiempo
list_voltaje = []
Tiempo = []
```

- Se utiliza **st.title** para agregar un título a la plataforma
- Luego se crean dos listas para almacenar los datos de voltaje y del tiempo para los intervalos

```
# Crear datos
Slider_Voltaje = st.slider("Voltaje (V)", 5, 50, 10, 1)

for Valores_voltaje in range(Slider_Voltaje):
    Valores_voltaje = rd.randint(5, 100)
    list_voltaje.append(Valores_voltaje)

# Crear los intervalos de tiempo de 0 a 10 segundos
Tiempo = np.linspace(0, 10, Slider_Voltaje)
```

- Le asignamos a la variable **Slider\_Voltaje** un slider en la plataforma que contiene los valores de valor mínimo el cual es 5, valor máximo el cual es 50, y valor de inicio el cual es 10, y un paso a paso de 1
- Luego se crea un for en rango del valor que se asigne en el slider, y dependiendo el numero que se elija, se asigna un valor aleatorio a la variable **Valores\_Voltaje**, y posteriormente se pasa a la lista creada anteriormente como **list\_voltaje**
- Con la variable **Tiempo** se crea el intervalo de 0 a 10, dependiendo cuantos voltajes escojamos.

```
# Crear un DataFrame para almacenar los datos
data = {"Tiempo (s)": Tiempo, "Voltaje (V)" : list_voltaje}
df = pd.DataFrame(data)

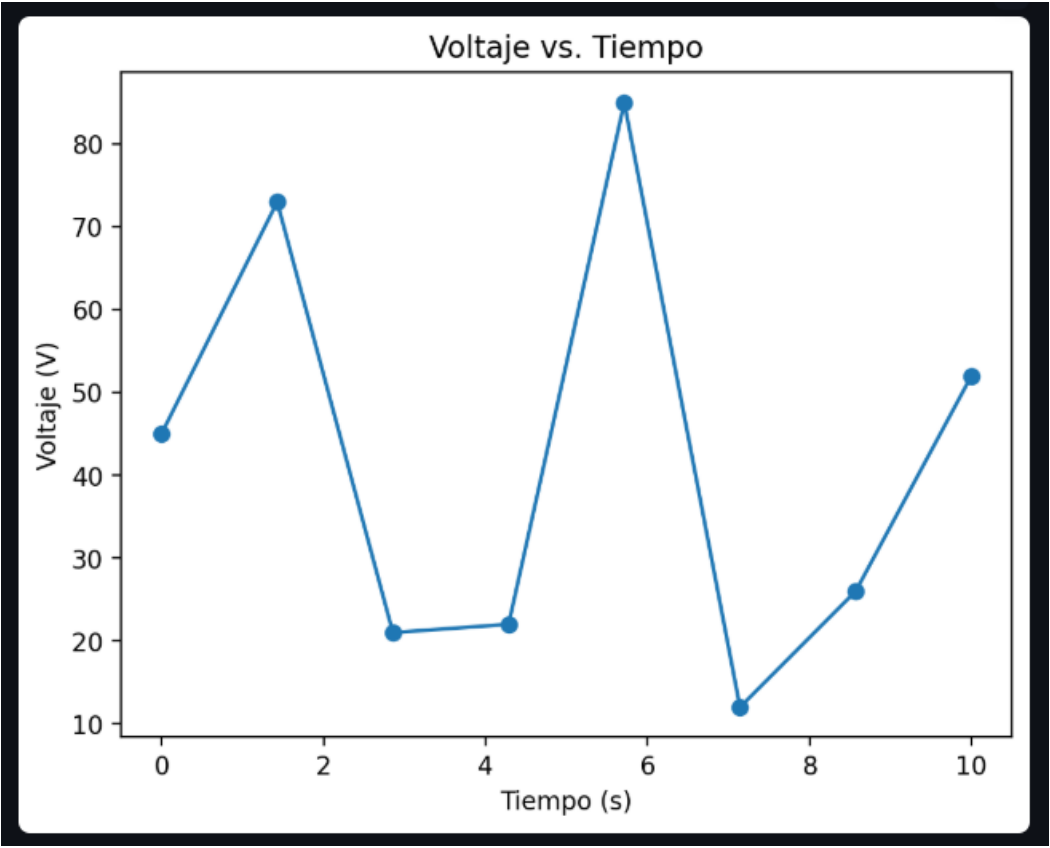
# Mostrar datos en una tabla
st.write("Datos de medición:")
st.dataframe(df)
```

- Se crea un Dataframe para almacenar los datos de tiempo y el voltaje
- Luego mostramos los anteriores datos en una tabla para visualizar en la plataforma.

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

- Con estas líneas de código se procede a graficar los datos de tiempo y voltaje.
- Se crea una figura donde en los valores de X iría el tiempo, y en los de Y iría el voltaje aleatorio.
- El título de la grafica seria Voltaje vs. Tiempo.
- Luego se muestra la figura en la plataforma.

EJECUCION DE STREAMLIT



## CONCLUSIONES

La realización de este proyecto permitió desarrollar con éxito una aplicación interactiva en Streamlit para la visualización de datos de voltaje contra tiempo. El proceso evidenció la importancia fundamental de definir claramente los requisitos funcionales, especialmente en elementos interactivos como los sliders, cuya interpretación (controlar cantidad de datos) impacta demasiado en la implementación. Se exploraron diversas técnicas para manejar la su alineación temporal para la graficación con Matplotlib y Pandas, incluyendo el uso de `numpy.linspace`, cada uno con sus ventajas en términos de precisión, simplicidad y dependencias.

## BIBLIOGRAFIA

- **Interactive Chaos. (s.f.). *Las funciones linspace y logspace*. Tutorial de NumPy. de <https://interactivechaos.com/es/manual/tutorial-de-numpy/las-funciones-linspace-y-logspace>**
- **Streamlit. (s.f.). *st.slider*. Streamlit Documentation: API reference. de <https://docs.streamlit.io/develop/api-reference/widgets/st.slider>**