CORPORACION UNIVERSITARIA AUTÒNOMA DEL CAUCA

AI EN LA WEB

ENTREGABLE Nº5 TALLER

CRISTIAN CAMILO MARTINEZ CORDOBA

ELECTIVA II

INGENIERÌA ELECTRONICA

ABRIL 2025

TALLER.

1. Modificar la aplicación para que los datos se generen aleatoriamente cada vez que se ejecuta.
2. Agregar un slider para permitir al usuario elegir el número de datos (es decir el número de registros de voltaje (con min_value=5, max_value=50 y el valor por defecto value=10), el rango de voltaje es libre y tiempo entre (0 - 10 segundos)).

Entregables: Documento en formato .pdf (su_nombre.pdf) con:

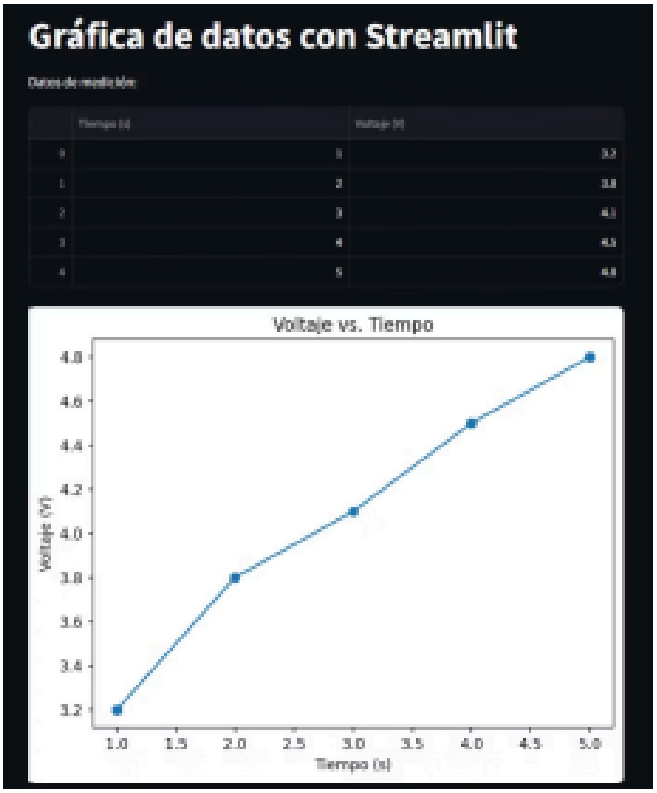
- Portada.1.
- Explicación del paso a paso con captura de pantalla del código y su ejecución desde el navegador web.2.
- Enlace del repositorio público en GitHub.3.
- Bibliografía.

DESARROLLO.

El código que se tiene como base para realizar este taller es el siguiente:

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
st.title("Gráfica de datos con Streamlit")
# Crear datos de ejemplo
data = {"Tiempo (s)": [1, 2, 3, 4, 5], "Voltaje (V)": [3.2, 3.8, 4.1, 4.5, 4.8]}
df = pd.DataFrame(data)
# Mostrar datos en una tabla
st.write("Datos de medición:")
st.dataframe(df)
# 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)
```

Obteniendo como resultado en el navegador lo siguiente:



Posterior a lo presentado anteriormente se procede a hacer un paso a paso de lo que se implementó en el código para dar solución a lo estipulado en el taller. Antes que nada, cabe definir un poco de las librerías que van a hacer posible la solución del taller: Primero, streamlit se importa como st, y es la herramienta principal que permite crear la interfaz web de forma sencilla y rápida. Con Streamlit se construyen los elementos visuales como el título, el control deslizante, la tabla de datos y la gráfica. En segundo lugar, se importa pandas como pd, una librería muy útil para el manejo de datos en forma de tablas, conocidas como DataFrames. En este caso, se utiliza para almacenar y mostrar los datos generados de tiempo y voltaje. Por último, matplotlib.pyplot se importa como plt, y se encarga de generar la gráfica de voltaje versus tiempo. Esta librería permite una personalización detallada de los gráficos, como los títulos, etiquetas y estilo de la línea.

Ahora bien, como primer paso se agregó un slider que permite al usuario elegir cuántos datos desea generar, esto se hace posible con la siguiente línea:

```
num_datos = st.slider("Selecciona la cantidad de datos a generar", min_value=5, max_value=50, value=10)
```

Esta línea crea una barra deslizante en la interfaz que va de 5 a 50, con un valor por defecto de 10. Gracias a esta línea, el usuario puede decidir cuántos puntos desea ver tanto en la tabla como en la gráfica, haciendo la app mucho más flexible e interactiva. Como segundo paso se cambiaron los datos fijos por datos aleatorios. En lugar de usar una lista escrita a mano como esta en la primera versión, ahora se generan los datos dinámicamente con NumPy. Esto se hace con estas dos líneas:

```
tiempo = np.linspace(0, 10, num_datos)

voltaje = np.random.uniform(low=0.5, high=5.0, size=num_datos)
```

La primera línea (np.linspace) genera una lista de valores entre 0 y 10 segundos, dependiendo de la cantidad seleccionada con el slider. La segunda (np.random.uniform) genera valores aleatorios

de voltaje dentro del rango 0.5 a 5.0, uno por cada instante de tiempo, así cada vez que se recarga la app, los datos cambian, simulando un nuevo experimento o medición. Luego, como tercer paso esos datos generados se agrupan en una tabla usando esta línea:

```
df = pd.DataFrame({"Tiempo (s)": tiempo, "Voltaje (V)": voltaje})
```

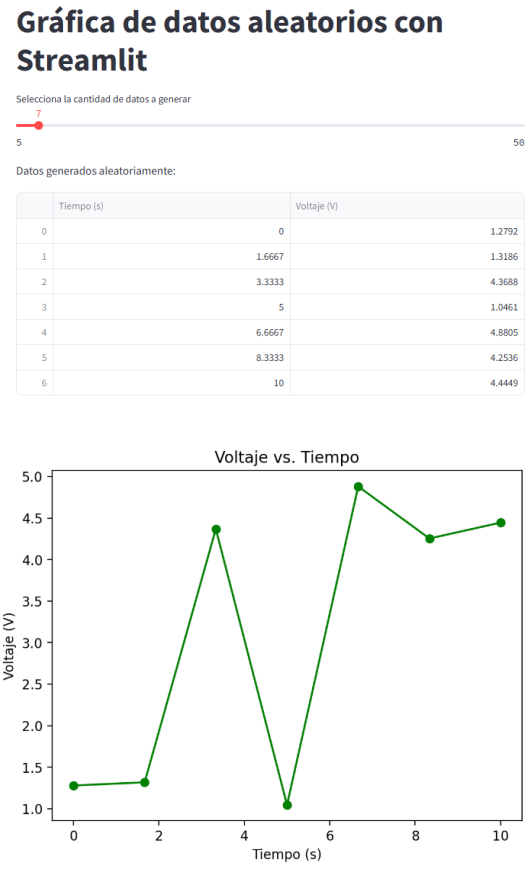
Esto crea una tabla con dos columnas: una para el tiempo y otra para el voltaje. Esta tabla se muestra en la app con la siguiente línea:

```
st.dataframe(df)
```

Gracias a esto, los usuarios pueden ver claramente los datos generados aleatoriamente antes de que se grafican. Como último paso se grafica la relación entre el voltaje y el tiempo utilizando Matplotlib, igual que en la versión original:

```
fig, ax = plt.subplots()

ax.plot(df["Tiempo (s)"], df["Voltaje (V)"], marker="o", linestyle="-", color="green")

ax.set_xlabel("Tiempo (s)")

ax.set_ylabel("Voltaje (V)")

ax.set_title("Voltaje vs. Tiempo")

st.pyplot(fig)
```

Estas líneas crean la figura, trazan los datos con líneas verdes y puntos redondos, y etiquetan los ejes y el título para que la gráfica sea clara. Finalmente, st.pyplot(fig) muestra la gráfica en la aplicación quedando de la siguiente manera:



Como se puede ver en el resultado final, todo lo explicado en el paso a paso se reflejó correctamente en la aplicación. El control deslizante permite al usuario definir la cantidad de datos a generar, y cada vez que se actualiza la app, se crean nuevos valores aleatorios de voltaje dentro del rango establecido. Estos datos se presentan tanto en una tabla como en una gráfica, lo que facilita su análisis visual.

ANEXOS

Código completo implementado:

```
import streamlit as st

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

# Slider para elegir el número de datos

num_datos = st.slider("Selecciona la cantidad de datos a generar",min_value=5,max_value=50,value=10)

# Generar datos aleatorios

tiempo = np.linspace(0, 10, num_datos)

voltaje = np.random.uniform(low=0.5, high=5.0, size=num_datos)

# Crear DataFrame

df = pd.DataFrame({

    "Tiempo (s)": tiempo,

    "Voltaje (V)": voltaje

})

# Mostrar datos

st.write("Datos generados aleatoriamente:")

st.dataframe(df)

# Graficar

fig, ax = plt.subplots()
```

```
ax.plot(df["Tiempo (s)"], df["Voltaje (V)"], marker="o", linestyle="-", color="green")

ax.set_xlabel("Tiempo (s)")

ax.set_ylabel("Voltaje (V)")

ax.set_title("Voltaje vs. Tiempo")

st.pyplot(fig)
```

REFERENCIAS.

https://moodle.uniautonoma.edu.co/pluginfile.php/224212/mod_assign/introattachment/0/Implementacion-de-IA-en-la-web.pdf?forcedownload=1

https://www.projectpro.io/recipes/add-slider-streamlit