



**UNIVO**  
**UNIVERSIDAD DE ORIENTE**  
EL SALVADOR, C.A.

Alumno:

MIGUEL DE JESÚS MIRANDA MEJÍA

Código Estudiantil:

(U20220148)

Docente:

EDUARDO JOSÉ VÁSQUEZ FLORES

Materia:

ESTRUCTURA DE DATOS SECCIÓN A

Trabajo:

Manual de elaboración de gráficos con  
Matplotlib

<b>CONTENIDO</b>	
INTRODUCCION .....	2
Objetivo General: .....	3
Objetivos Específicos: .....	3
¿Qué es matplotlib? .....	4
Características: .....	4
Áreas de implementación de Matplotlib .....	4
Ejemplos de Gráficos con Código: .....	5
Histograma: .....	5
Diagrama de Barras: .....	6
Gráfico de Pastel:.....	6
Gráfico de Dispersión: .....	7
Alternativas a Matplotlib: .....	8
CONCLUSION .....	9
Bibliografía .....	10

## INTRODUCCION

En el mundo contemporáneo, la competencia en el análisis de datos se ha convertido en una destreza crítica en una amplia gama de disciplinas. La visualización de datos juega un papel esencial en este proceso, ya que facilita la interpretación rápida y clara de patrones y tendencias. En este contexto, la biblioteca de Python, Matplotlib, ha demostrado ser una herramienta excepcionalmente potente para la creación de gráficos que son tanto informativos como atractivos. Este estudio se centra en una exploración exhaustiva de Matplotlib, desde sus conceptos básicos hasta su aplicación práctica en diversos tipos de gráficos. Al dominar esta herramienta, se fortalecen las habilidades necesarias para representar datos de manera efectiva y comunicar descubrimientos de forma precisa y convincente.

## Objetivo General:

Desarrollar un profundo conocimiento sobre la biblioteca Matplotlib y sus aplicaciones en visualización de datos, con el fin de crear visualizaciones precisas y efectivas para analizar y comunicar información relevante en diversos contextos.

## Objetivos Específicos:

1. **Desarrollar Proficiencia Técnica:** Adquirir un dominio completo de los principios esenciales de Matplotlib, lo que abarca su sintaxis, variedad de gráficos disponibles y técnicas de personalización. Esto permitirá la creación de visualizaciones de datos complejas y con un significado claro.
2. **Implementar Conceptos en Situaciones Prácticas:** Utilizar los conocimientos obtenidos en proyectos concretos para generar al menos cinco tipos distintos de gráficos, como gráficos de líneas, de barras, de dispersión, de pastel e histogramas, empleando Matplotlib. Se busca comprender el código subyacente de cada gráfico y apreciar su pertinencia en contextos específicos de análisis de datos.

## ¿Qué es matplotlib?

Se trata de una librería que posibilita la representación visual de datos en dos y tres dimensiones dentro del lenguaje de programación Python. Facilita la creación de una diversidad de gráficos, que pueden ser estáticos, interactivos o incluso animados.

## Características:

**Flexibilidad:** Matplotlib proporciona una amplia gama de gráficos y opciones de personalización.

**Simplicidad:** Su sintaxis es simple y fácil de comprender.

**Compatibilidad:** Es compatible con varios entornos y sistemas operativos.

**Comunidad Activa:** Cuenta con una comunidad activa que ofrece tutoriales y soluciones para desafíos comunes.

## Áreas de implementación de Matplotlib

**Científico:** En el ámbito científico, se utiliza para representar visualmente los resultados de experimentos y datos científicos.

**Finanzas:** En finanzas, se emplea para graficar tendencias del mercado y llevar a cabo análisis financiero.

**Biológico:** En biología y medicina, se utiliza para visualizar datos biológicos y médicos.

**Ingeniería:** En ingeniería, se emplea para presentar datos de ingeniería y modelado.

**Educación:** En el ámbito educativo, se utiliza para enseñar conceptos matemáticos y estadísticos de forma visual.

## Ejemplos de Gráficos con Código:

### Gráfico en Línea:

**Uso:** Para representar tendencias a lo largo del tiempo.

#### Ejemplo:

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

plt.plot(x, y)
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.title('Gráfico de Línea')
plt.show()
```

### Histograma:

**Uso:** Para mostrar la distribución de datos numéricos.

#### Ejemplo:

```
import matplotlib.pyplot as plt

data = [1, 2, 2, 3, 3, 3, 4, 4, 5, 5]

plt.hist(data, bins=5, edgecolor='black')
plt.xlabel('Valores')
plt.ylabel('Frecuencia')
plt.title('Histograma')
plt.show()
```

## Diagrama de Barras:

**Uso:** Para comparar diferentes categorías.

### Ejemplo:

```
import matplotlib.pyplot as plt

categorias = ['A', 'B', 'C', 'D', 'E']
valores = [4, 7, 1, 6, 2]

plt.bar(categorias, valores)
plt.xlabel('Categorías')
plt.ylabel('Valores')
plt.title('Diagrama de Barras')
plt.show()
```

## Gráfico de Pastel:

**Uso:** Para mostrar proporciones de un todo.

### Ejemplo:

```
import matplotlib.pyplot as plt

etiquetas = ['A', 'B', 'C', 'D']
tamaños = [25, 30, 15, 30]
colores = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']

plt.pie(tamaños, labels=etiquetas, colors=colores, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title('Gráfico de Pastel')
plt.show()
```

## Gráfico de Dispersión:

**Uso:** Para visualizar la relación entre dos variables.

### Ejemplo:

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 4, 5]
y = [2, 3, 5, 6, 4, 6]

plt.scatter(x, y, color='blue')
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.title('Gráfico de Dispersión')
plt.show()
```

## Alternativas a Matplotlib:

1. Seaborn, que tiene sus raíces en Matplotlib, provee una interfaz de nivel superior para la creación de gráficos estadísticos atractivos y llenos de información.
2. Plotly destaca por sus gráficos interactivos, siendo especialmente valioso para visualizaciones en entornos web.
3. Bokeh se presenta como la elección ideal para generar gráficos interactivos en tiempo real.
4. Altair, caracterizada como una biblioteca declarativa, facilita la creación de visualizaciones simples pero altamente efectivas.
5. Por último, ggplot, basada en el paquete ggplot2 de R, ofrece una sintaxis análoga para la generación de gráficos gramaticales en Python.

## CONCLUSION

En resumen, este análisis detallado de Matplotlib ha brindado una comprensión práctica y exhaustiva sobre cómo visualizar datos en Python. Mediante la exploración de sus múltiples capacidades y variedad de gráficos, hemos desarrollado habilidades fundamentales para representar información compleja de manera clara y comprensible. La habilidad para elegir y generar visualizaciones apropiadas de acuerdo al contexto del análisis de datos se revela como crucial en la toma de decisiones bien fundamentadas.

## Bibliografía

<https://datascientest.com/es/todo-sobre-matplotlib>

[API Reference — Matplotlib 3.8.0 documentation](#)

<https://www.plandemejora.com/para-que-sirve-la-grafica-de-barras/>

<https://datos.gob.es/es/blog/11-librerias-para-crear-visualizaciones-de-datos>

[Matplotlib — Visualization with Python](#)

[Data Science Projects with Python: A case study approach to gaining valuable ...](#)

[- Stephen Klosterman - Google Libros](#)