

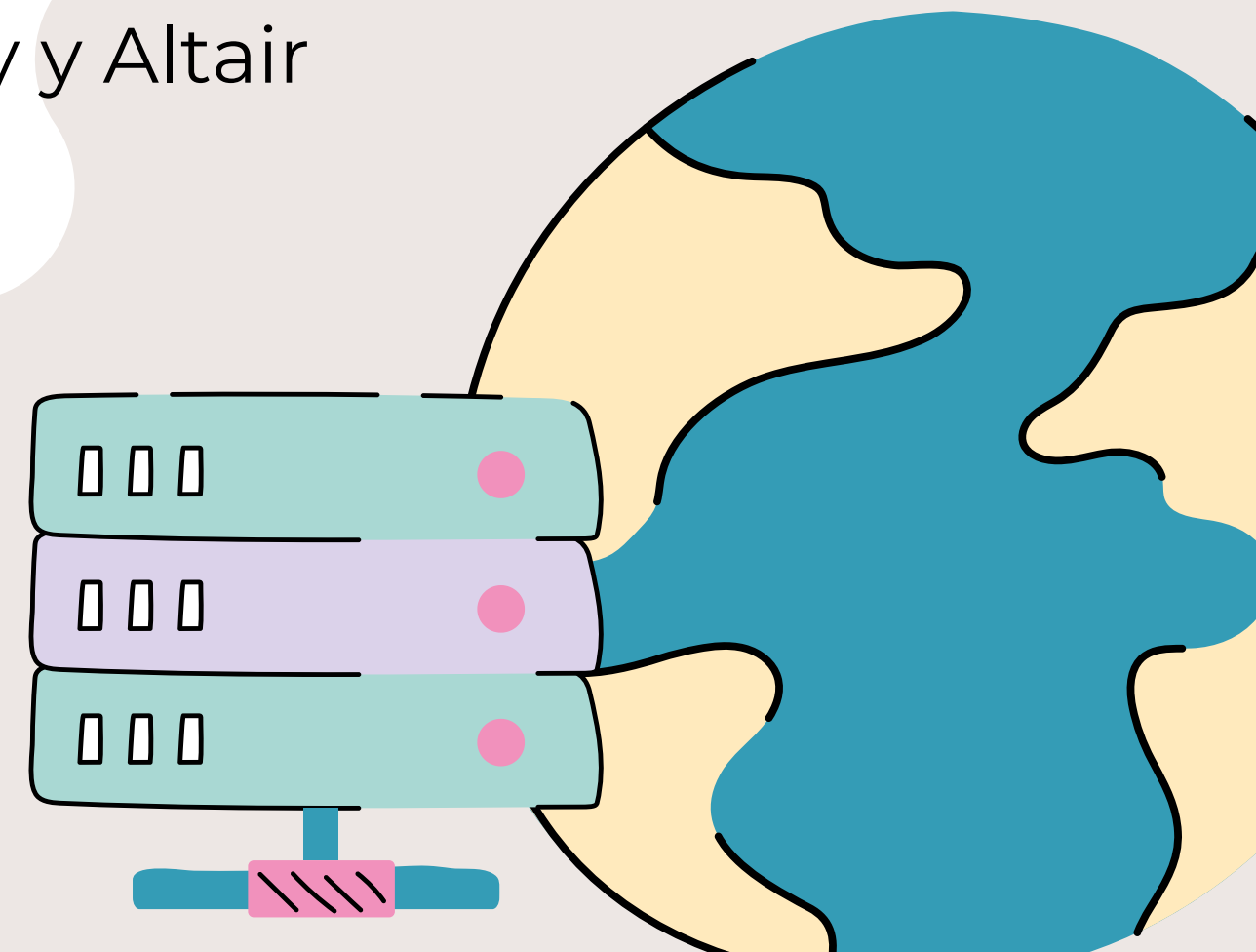


# Visualización de datos con Python

Comparando Matplotlib, Seaborn, Plotly y Altair



Por Juan Duran



# Introducción

La **visualización** de **datos** es una de las **herramientas** más poderosas para comprender y **comunicar información**. Un buen gráfico puede revelar **tendencias** ocultas, facilitar la toma de **decisiones** y hacer que los datos sean mucho más accesibles.

En Python, existen diversas **librerías** especializadas en visualización. Pero, ¿cómo elegir la mejor para cada caso? En esta presentación, compararemos cuatro opciones populares: **Matplotlib**, **Seaborn**, **Plotly** y **Altair**. Además, exploraremos algunas **mejores prácticas** para crear visualizaciones impactantes y efectivas.



# Pros

- Versatilidad
- Integración con herramientas de análisis
- Personalización avanzada
- Capacidad para manejar grandes volúmenes de datos
- Comunidad y documentación extensa
- Opciones interactivas



**versus**

# Contras

- Curva de aprendizaje
- Dificultad en personalización avanzada
- Rendimiento en grandes datasets
- Interactividad limitada
- Compatibilidad
- Dependencia de configuración del entorno

# Puntos clave



## Elección librería

No todas sirven para lo mismo. Hay que **elegir** según el tipo de gráfico y el nivel de interactividad deseado.



## Claridad y simplicidad

Un gráfico recargado puede ser más confuso que útil. Hay que **evitar** el exceso de elementos innecesarios.



## Uso color y etiquetas

Los colores deben **ayudar** a diferenciar información, no distraer. Las etiquetas y títulos deben ser claros.

# Matplotlib

**Ideal** para: **Gráficos estáticos** y **personalización** detallada. Es perfecto cuando se necesita control total sobre los detalles del gráfico, como colores, fuentes, tamaños, etiquetas y leyendas. Es ampliamente utilizado en entornos de producción donde la **consistencia** y **calidad** del gráfico son cruciales.

## Pros:

- Totalmente personalizable.
- Compatible con otras herramientas.
- Ideal para producción.

## Contras:

- Sintaxis compleja.
- No es interactivo por defecto.

**Personalización**

**Gráficos estáticos**

**Control total**

**Producción**



# Seaborn

**Ideal** para: **Visualizaciones rápidas y atractivas**, especialmente cuando se trabaja con **datos estadísticos**. Si el objetivo es obtener una visualización rápida y comprensible sin mucha personalización detallada, Seaborn es una excelente opción.

## Pros:

- Estilo predeterminado atractivo.
- Integración fácil con Pandas.
- Simplifica Matplotlib.

## Contras:

- Menos flexible que Matplotlib.
- Menos opciones de interactividad.

**Estética**

**Estadística**

**Pandas**

**Simplicidad**



# Plotly

**Ideal** para: **Visualizaciones interactivas** y **dashboards**. Si tu objetivo es crear gráficos interactivos que permitan a los usuarios explorar los datos, Plotly es una opción líder.

## Pros:

- Soporte para zoom y hover.
- Integración con web y dashboards.

## Contras:

- Sintaxis diferente a Matplotlib.
- Requiere instalación adicional.

**Dashboards**

**Interactividad**



**Zoom**

**Web**



# Altair

**Ideal** para: **Gráficos declarativos** y **rápidos**.

Altair es ideal cuando se quiere crear gráficos rápidamente a partir de datos estructurados y cuando se desea un enfoque más simple pero eficiente para prototipos.

## Pros:

- Fácil de usar.
- Basada en una sintaxis declarativa.
- Ideal para prototipos rápidos.

## Contras:

- Menos flexible para personalización avanzada.
- Menos documentación que otras.

**Rápido**

**Declarativo**



**Prototipo**

**Sintaxis sencilla**







# Mejores prácticas



## **Menos es más**

Evita el exceso de elementos que distraigan del mensaje principal.

## **Colores con intención**

No abuses de demasiados colores sin significado.

## **Elige gráfico adecuado**

No todos los datos se representan igual de bien en todos los formatos.

## **Gráfico legible**

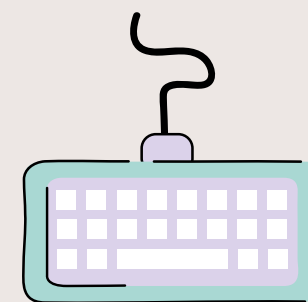
Escoge bien las escalas, etiquetas y tamaños.

## **Coherencia visual**

Usa estilos similares en todas tus visualizaciones.

## **Historia con los datos**

El gráfico debe responder preguntas, no solo mostrar números.



# Gracias



Por Juan Duran

**“Coding, Gaming and Leveling Up”**