

«Talento Tech»

Data Analytics

con Python

Clase 15



Clase 15 | Visualización Interactiva con Plotly

Temario:

- Introducción a la visualización interactiva.
 - Comparación con Matplotlib y Seaborn.
 - Creación de gráficos dinámicos y dashboards.
-

Objetivos de la Clase:

- Entender qué son las visualizaciones interactivas y por qué son importantes en el análisis de datos moderno.
- Presentar Plotly como una librería de Python para crear visualizaciones interactivas.
- Comparar Plotly con otras herramientas.

Visualizaciones Interactivas con Plotly

¿Qué implica una Visualización Interactiva?

En el ámbito del análisis de datos, no es suficiente presentar un gráfico estático. Se vuelve necesario **permitir que los usuarios interactúen con los datos**: explorar, examinar y manipular la información. Las visualizaciones interactivas cumplen esta función, ofreciendo gráficos que responden a las acciones del usuario. Funcionalidades como hacer zoom, visualizar valores exactos al pasar el cursor, o filtrar datos, facilitan y enriquecen la comprensión de la información.



Plotly: Una Herramienta Innovadora

Plotly es una librería de Python diseñada para crear visualizaciones interactivas de manera accesible. Permite generar gráficos tanto en línea, facilitando su compartición, como fuera de línea, brindando flexibilidad en su uso.

Comparativa: Plotly frente a Matplotlib y Seaborn

Matplotlib	Considerada la base de las visualizaciones en Python, destaca por su solidez y fiabilidad, aunque se limita a gráficos estáticos con escasa interactividad.
Seaborn	Extensión de Matplotlib, simplifica la creación de gráficos estadísticos más estéticos y complejos. Sin embargo, comparte la limitación de interactividad presente en Matplotlib.
Plotly	Ofrece interactividad inherente a sus gráficos, permitiendo la exploración y manipulación directa de los datos. Ideal para presentaciones que buscan involucrar al público.

Gráficos Comunes en Plotly y su Personalización

Plotly proporciona una amplia variedad de opciones gráficas. A continuación, se presentan ejemplos con código para facilitar la experimentación:

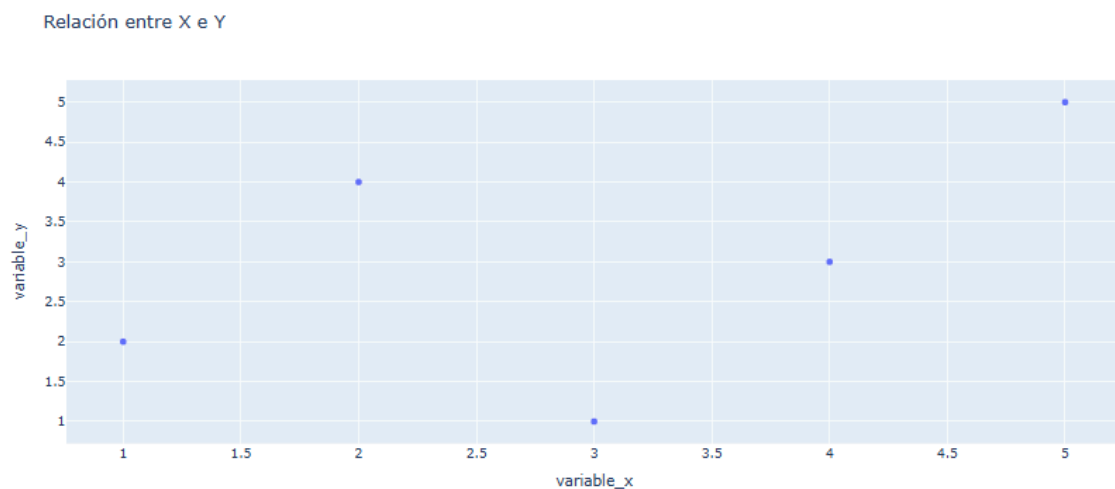
Gráfico de Dispersión (Scatter Plot):

Visualizar la relación entre dos variables.

```
import plotly.express as px
import pandas as pd

# Asumiendo un DataFrame 'df' con columnas 'variable_x' y 'variable_y'
data = {'variable_x': [1, 2, 3, 4, 5],
        'variable_y': [2, 4, 1, 3, 5]}
df = pd.DataFrame(data)

fig = px.scatter(df, x="variable_x", y="variable_y",
title="Relación entre X e Y")
fig.show()
```



Personalización: Modificar colores, tamaños de puntos y añadir etiquetas según necesidad.

```
fig = px.scatter(df, x="variable_x", y="variable_y",
                 color="variable_y", size="variable_x",
                 hover_data=['variable_x', 'variable_y'])
fig.update_layout(title="Relación entre X e Y (Personalizado)")
fig.show()
```

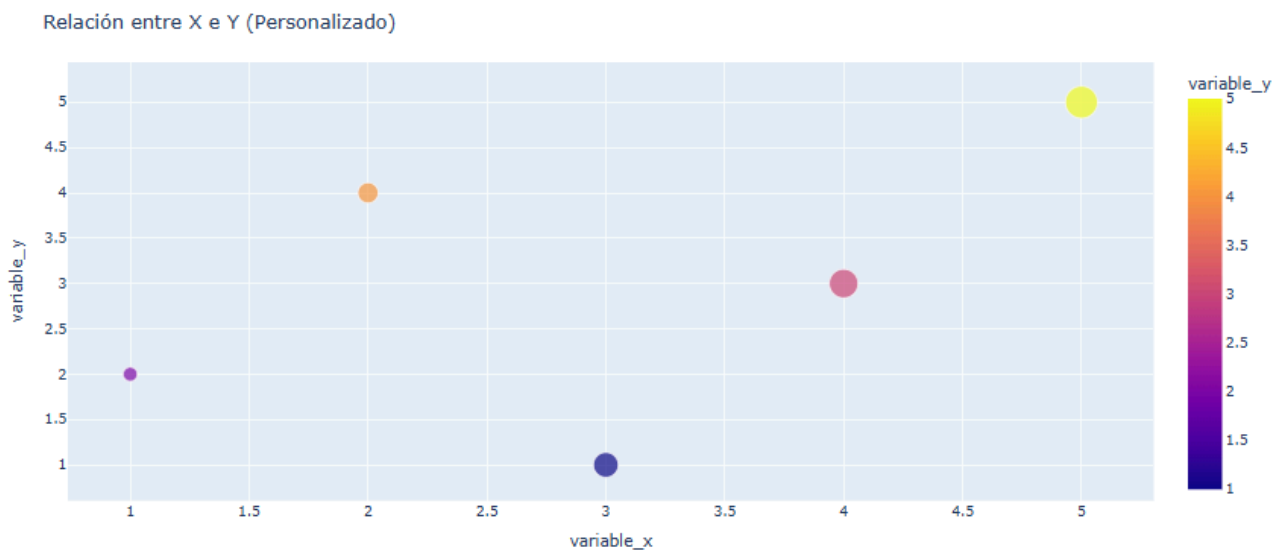


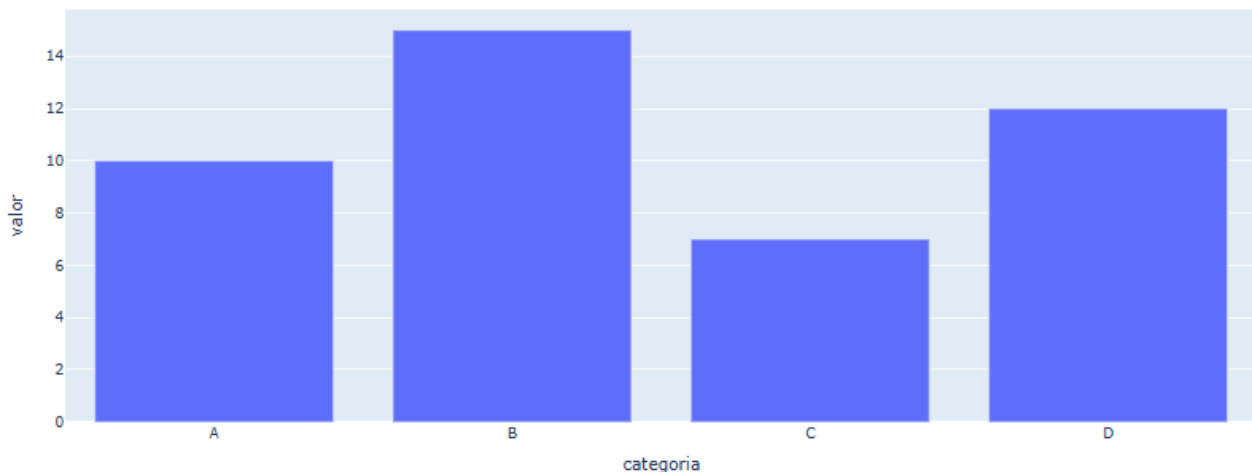
Gráfico de Barras (Bar Chart):

Comparar diferentes categorías.

```
data = {'categoria': ['A', 'B', 'C', 'D'],
        'valor': [10, 15, 7, 12]}
df = pd.DataFrame(data)

fig = px.bar(df, x="categoria", y="valor", title="Comparación
de Categorías")
fig.show()
```

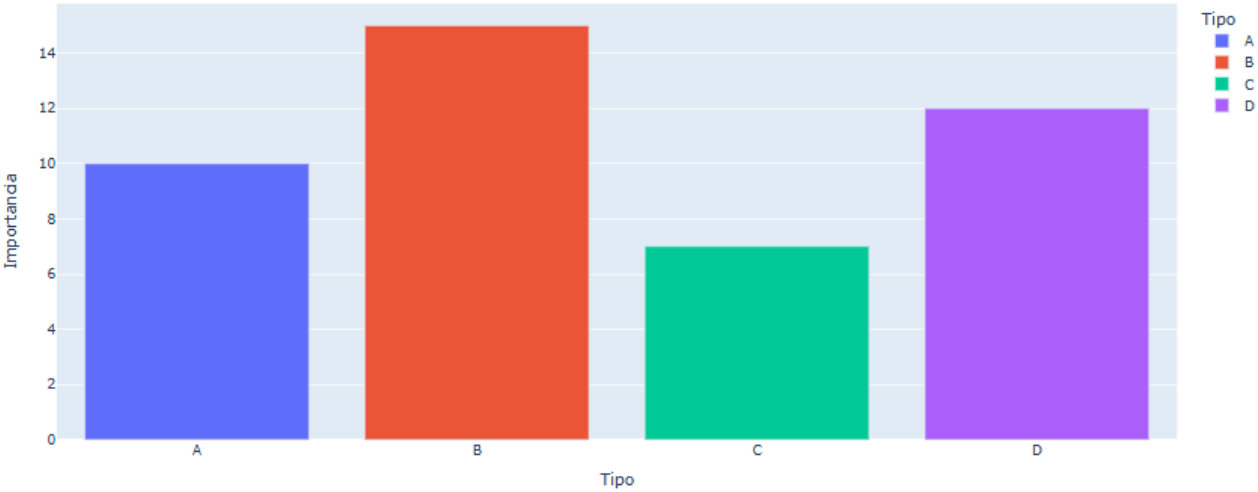
Comparación de Categorías



Personalización: Ajustar colores, añadir etiquetas y orientar las barras horizontalmente, entre otras opciones.

```
fig = px.bar(df, x="categoria", y="valor", color="categoria",
             hover_data=['valor'],
             labels={'valor': 'Importancia',
                    'categoria': 'Tipo'}) # Etiquetado personalizado
fig.update_layout(title="Comparación de Categorías
(Personalizado)")
fig.show()
```

Comparación de Categorías (Personalizado)

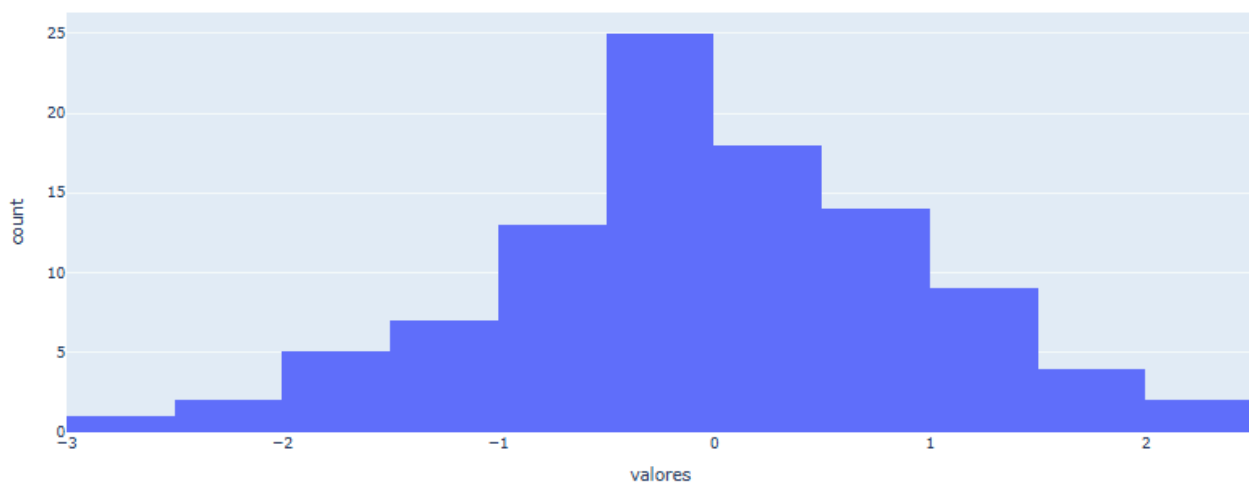


Histograma:

Analizar la distribución de una variable.

```
import numpy as np
data = {'valores': np.random.randn(100)}
df = pd.DataFrame(data)

fig = px.histogram(df, x="valores", title="Distribución de Valores")
fig.show()
```



Personalización: Definir el número de bins, cambiar colores y superponer una curva de densidad.

```
fig = px.histogram(df, x="valores", nbins=30,
color_discrete_sequence=['#EB89B5'])
fig.update_layout(title="Distribución de Valores (Personalizado)")
fig.show()
```

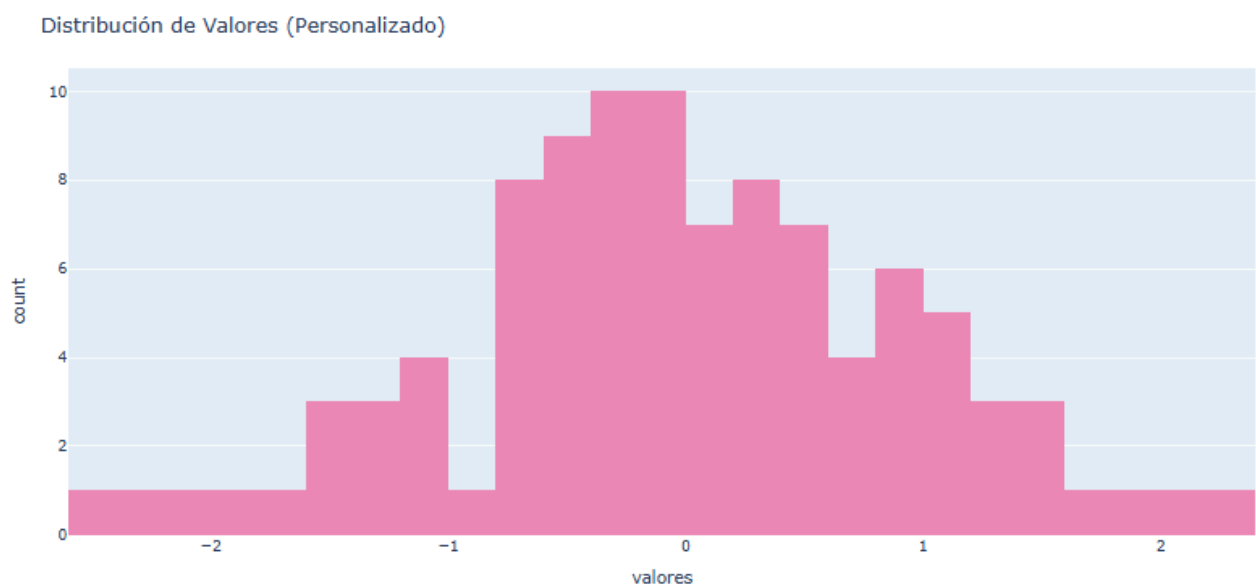
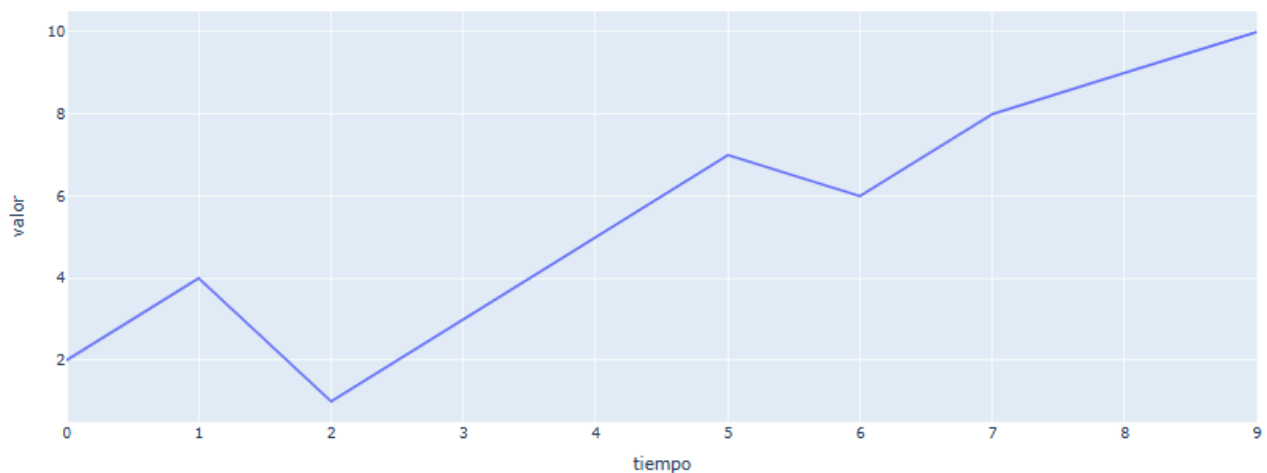


Gráfico de Líneas (Line Chart):

Observar la evolución de una variable a lo largo del tiempo.

```
data = {'tiempo': range(10),  
        'valor': [2, 4, 1, 3, 5, 7, 6, 8, 9, 10]}  
df = pd.DataFrame(data)  
  
fig = px.line(df, x="tiempo", y="valor", title="Evolución en  
el Tiempo")  
fig.show()
```

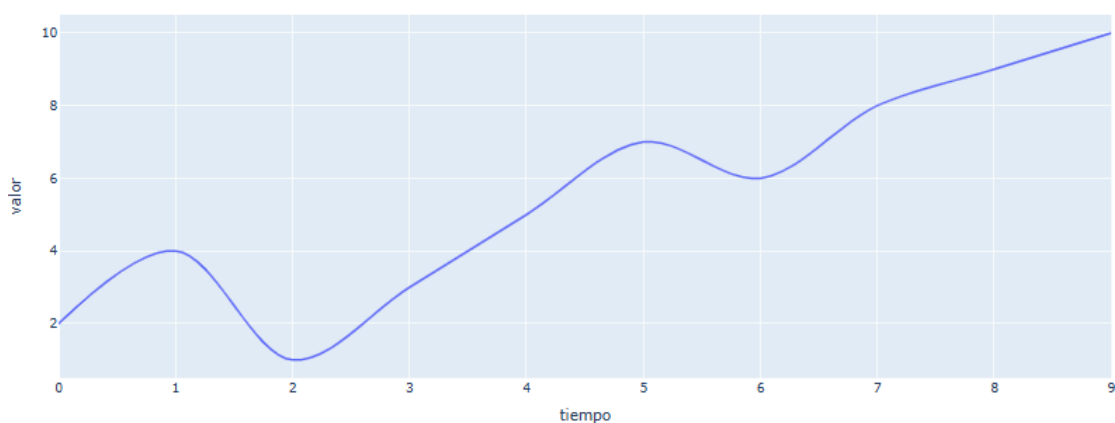
Evolución en el Tiempo



Personalización: Modificar el color de la línea, añadir marcadores y mostrar rangos de confianza.

```
fig = px.line(df, x="tiempo", y="valor", title="Evolución en  
el Tiempo (Personalizado)",  
              line_shape="spline", render_mode="svg") # Line  
shape suavizada  
fig.show()
```

Evolución en el Tiempo (Personalizado)



Crear un Dashboard Sencillo:

Un Centro de Control de Datos

Un dashboard integra múltiples gráficos y controles interactivos en un mismo espacio. Dash, una herramienta de Plotly, simplifica el proceso de creación de dashboards.

Ejemplo básico:

```
import dash
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
import pandas as pd

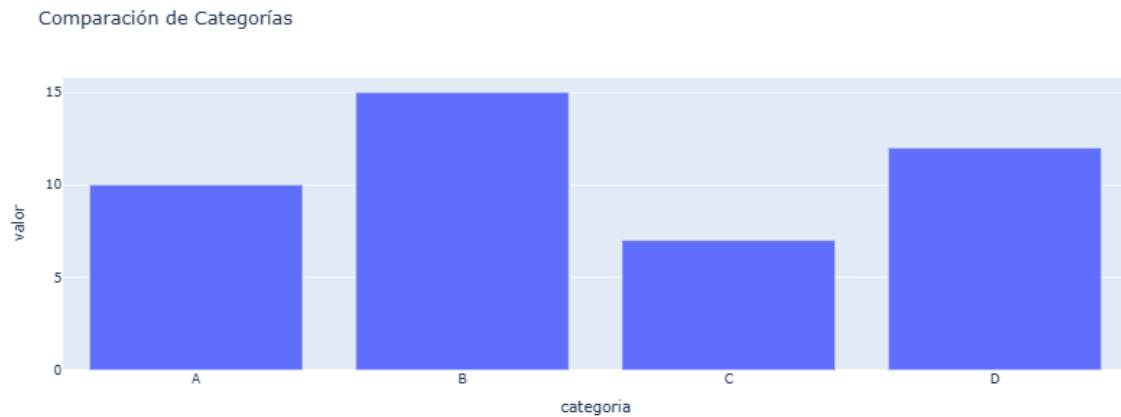
app = dash.Dash(__name__)
# Generar datos de ejemplo
data = {'categoria': ['A', 'B', 'C', 'D'],
        'valor': [10, 15, 7, 12]}
df = pd.DataFrame(data)
fig = px.bar(df, x="categoria", y="valor", title="Comparación
de Categorías")

app.layout = html.Div(children=[
    html.H1(children='Mi Primer Dashboard con Dash'),

    dcc.Graph(
        id='ejemplo-grafico',
        figure=fig
    )
])
if __name__ == '__main__':
    app.run_server(debug=True)
```

Este código genera un dashboard con un título y un gráfico de barras. Se puede ampliar con más gráficos y filtros.

Mi Primer Dashboard con Dash



Aplicaciones Prácticas: Ejemplos de Uso

- **Análisis de Ventas:** Visualizar la evolución de ventas por producto, región o vendedor. Permitir filtrar por fecha para analizar resultados específicos.
- **Análisis de Redes Sociales:** Mostrar la cantidad de seguidores, "me gusta" y comentarios a lo largo del tiempo. Facilitar el zoom en momentos de alta actividad.
- **Análisis de Encuestas:** Presentar las respuestas a las preguntas de la encuesta. Ofrecer la posibilidad de filtrar por edad, género o nivel educativo.
- **Monitoreo de Indicadores Clave (KPIs):** Desarrollar un dashboard con los KPIs más relevantes. Permitir visualizar el estado actual, la tendencia y los objetivos.



Reflexión final

Plotly es una herramienta valiosa para crear visualizaciones interactivas que aportan dinamismo a los datos. En comparación con Matplotlib y Seaborn, facilita una **experiencia más interactiva e interesante para el usuario**. Se recomienda explorar sus diversas opciones para crear dashboards que mejoren la comprensión de la información.

Materiales y recursos adicionales

- [Plotly](#)
- [Dash](#)

Próximos Pasos

- El próximo paso: El científico de datos. Machine Learning. Demostración de regresión lineal
- Tips de inserción laboral.
- Cierre de cursada

Ejercicios Prácticos



DATAWISE CONSULTING
INSIGHTS ESTRATÉGICOS

Actividad 1: Análisis de Ventas con Gráficos Interactivos

Contexto:



Durante esta semana de tu pasantía, tu primera tarea consiste en analizar datos de ventas. Silvia, Project Manager y Data Scientist, requiere un análisis detallado del rendimiento de las ventas de los productos principales.

Objetivos:

- Crear visualizaciones interactivas para analizar las ventas de diferentes productos.
- Identificar tendencias y patrones clave en los datos de ventas.
- Presentar los hallazgos de manera clara y concisa utilizando Plotly.

Ejercicio práctico:

1. Se solicita la creación de gráficos interactivos que permitan visualizar la evolución de las ventas a lo largo del tiempo, comparar el rendimiento de los diferentes productos y filtrar por región.
2. **Desarrollo:**
 - Cargar el dataset de ventas
 - Utilizar Plotly para crear un gráfico de líneas que muestre la evolución de las ventas a lo largo del tiempo para cada producto.
 - Implementar interactividad: Permitir que los usuarios puedan pasar el cursor sobre las líneas para visualizar los valores exactos de ventas en cada fecha.
 - Crear un gráfico de barras que compare las ventas totales de cada producto.
 - Implementar filtros para que los usuarios puedan seleccionar una región específica y visualizar las ventas únicamente para esa región.
3. **Presentación:** Preparar una presentación concisa que incluya las visualizaciones y una explicación de los hallazgos principales.

Dataset (sample)

```
data = {'fecha': pd.to_datetime(['2024-01-01', '2024-01-01',  
'2024-01-08', '2024-01-08', '2024-01-15', '2024-01-15']),  
        'producto': ['A', 'B', 'A', 'B', 'A', 'B'],  
        'region': ['Norte', 'Sur', 'Norte', 'Sur', 'Norte', 'Sur'],  
        'ventas': [10, 15, 12, 18, 15, 20]}
```

¿Por qué importa esto en SynthData?

En SynthData, el análisis de ventas es fundamental para tomar decisiones informadas sobre el desarrollo de productos, las estrategias de marketing y la expansión a nuevas regiones. Una visualización interactiva permite al equipo explorar los datos en detalle, identificar oportunidades de crecimiento y anticipar posibles desafíos. El desarrollo de estas habilidades contribuye a la formación de analistas de datos competentes.

Actividad 2: Dashboard de Redes Sociales



Contexto:

Sabrina, la Data Engineer, ha implementado un sistema para recopilar datos de redes sociales. Se requiere el diseño de un dashboard para visualizar y analizar estos datos.

Objetivos:

- Crear un dashboard interactivo para mostrar las métricas clave de redes sociales.
- Permitir a los usuarios filtrar los datos por plataforma y período de tiempo.
- Diseñar un dashboard visualmente atractivo y fácil de usar.

Ejercicio práctico:

1. Se solicita la creación de un dashboard que muestre la cantidad de seguidores, "me gusta" y comentarios en las redes sociales de la empresa. Es importante poder visualizar la evolución de estas métricas a lo largo del tiempo y filtrar por plataforma para identificar áreas de mayor impacto.
2. **Desarrollo:**
Cargar el dataset de redes sociales
Utilizar Dash y Plotly para crear un dashboard con los siguientes elementos:


- Un gráfico de líneas que muestre la evolución de la cantidad de seguidores a lo largo del tiempo para cada plataforma (Facebook, Twitter, Instagram, etc.).
- Un gráfico de barras que compare la cantidad de "me gusta" y comentarios en cada plataforma.
- Filtros que permitan a los usuarios seleccionar una o varias plataformas y un período de tiempo específico.
- Títulos claros y concisos para cada gráfico.

Dataset (sample)

```
data = {'fecha': pd.to_datetime(['2024-01-01', '2024-01-01', '2024-01-08',
                                '2024-01-08', '2024-01-15', '2024-01-15']),
        'plataforma': ['Facebook', 'Twitter', 'Facebook', 'Twitter',
                        'Facebook', 'Twitter'],
        'seguidores': [1000, 500, 1100, 550, 1200, 600],
        'likes': [100, 50, 110, 55, 120, 60],
        'comentarios': [10, 5, 11, 5, 12, 6]}
```

¿Por qué importa esto en SynthData?

El monitoreo de las redes sociales es crucial para comprender el impacto de las campañas de marketing, identificar tendencias emergentes y mejorar la interacción con los clientes. Un dashboard interactivo permite al equipo tomar decisiones basadas en datos y optimizar sus estrategias en tiempo real. Esta actividad proporciona una visión integral del ciclo de vida de los datos, desde la recolección hasta la visualización y el análisis.

 **Estos ejercicios son una simulación de cómo se podría resolver el problema en este contexto específico. Las soluciones encontradas no aplican de ninguna manera a todos los casos.**

Recuerda que las soluciones dependen de los sets de datos, el contexto y los requerimientos específicos de los stakeholders y las organizaciones.



Buenos Aires
aprende
Agencia de Habilidades para el Futuro

BA Buenos
Aires
Ciudad