**Laboratorio 23**

# Sesión # 23 Graficas de Power BI

**Título del Laboratorio:** Aplicación y uso de las gráficas de los datos con Python (Gráficas más avanzadas con Python con la combinación de librerías)

**Duración:** 2 horas

**Objetivos del Laboratorio:** *Afianzar los conocimientos de las gráficas de los datos con Python (Gráficas más avanzadas con Python con la combinación de librerías) con escenarios prácticos.*

# Materiales Necesarios:

1. *Computador con acceso a internet.*
2. *Colocarlo en el repositorio de GitHub*
3. *Ampliar el conocimiento con el curso de datos en AWS y Cisco.*
4. *Python, Google colab*

# Estructura del Laboratorio:

**Parte 1**

En la primera parte se aplicarán los temas vistos en la sesión como las gráficas de los datos con Python (Gráficas más avanzadas con Python con la combinación de librerías), se deberá realizar el paso a paso con las respetivas capturas de pantalla, esta aplicación es de acuerdo con los escenarios planteados.

Realización de la unidad del curso de AWS o Cisco y anexar captura de pantalla del avance.

# Ejercicio de práctica 1.

Realizar el paso a paso de las gráficas en Python para los datos, deberás realizar las respectivas capturas de pantalla, conclusión, guardar el archivo.

# Escenario 1: Gráfico de Barras 2D en Diferentes Planos en 3D

Este ejercicio tiene como objetivo crear un gráfico 3D que proyecte gráficos de barras 2D en diferentes planos del espacio tridimensional. En este caso, se proyectan barras en los planos y=0, y=1, y=2, y y=3, lo que crea la ilusión de varios gráficos de barras organizados a lo largo de diferentes capas o planos.

np.random.seed(19680801)

# Tener en cuenta:

* + Visualizar gráficos de barras 2D en diferentes posiciones dentro de un espacio tridimensional.
  + Distribuir los gráficos de barras en diferentes planos en el eje y del gráfico, mostrando cómo se comportan los datos a lo largo de este eje.
  + Usar colores diferentes para cada conjunto de barras proyectadas, facilitando la diferenciación entre los distintos planos.
  + Crear una visualización en 3D que permita explorar la variación de las barras a lo largo de un eje adicional (en este caso, el eje y).

**Paso a paso:**

1. Importación de librerías: Usaremos numpy para generar los datos y matplotlib para la visualización 3D.

2. Generación de datos: Generaremos datos aleatorios para las barras, y asignaremos distintos valores y a las distintas capas del gráfico de barras.

3. Configuración del gráfico 3D: Usaremos la función Axes3D de matplotlib para crear un gráfico tridimensional.

4. Distribución de las barras en los planos: Proyectaremos distintos conjuntos de barras a distintos valores y, de forma que las barras de cada conjunto queden alineadas a lo largo del eje y.

5. Diferenciación por colores: Usaremos distintos colores para cada conjunto de barras, de forma que sea fácil diferenciarlas**.**

**CÓDIGO**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

# Escenario 2: Gráfico de Superficie 3D (Mapa de Colores)

Este ejercicio tiene como objetivo crear una superficie 3D utilizando Matplotlib y aplicar un mapa de colores que represente la variación de los valores en la superficie.

* El gráfico utiliza una función matemática para generar la superficie, y se colorea utilizando un gradiente de colores cálidos y fríos (rojo para valores altos y azul para valores bajos).
* El gráfico también incluye una barra de colores para interpretar visualmente los valores en la superficie.

Demuestra cómo trazar una superficie 3D coloreada con el mapa de colores frío y cálido. La superficie se vuelve opaca utilizando antialiased=False.

También demuestra el uso del LinearLocatorformato personalizado para las etiquetas de marca del eje z.

#datos.

X = np.arange(-5, 5, 0.25)

Y = np.arange(-5, 5, 0.25) X, Y = np.meshgrid(X, Y) R = np.sqrt(X\*\*2 + Y\*\*2) Z = np.sin(R)

# Tener en cuenta que:

* + Visualizar una superficie tridimensional generada por una función matemática.
  + Aplicar un mapa de colores que refleje los cambios en los valores de la superficie. En este caso, se utiliza un gradiente de azul a rojo (colores fríos a cálidos), donde los valores más altos están en rojo y los más bajos en azul.
  + Controlar la apariencia de la superficie haciéndola opaca mediante el uso de antialiased=False, lo que evita suavizar los bordes de la malla, dándole un aspecto más nítido.
  + Personalizar las etiquetas del eje z usando LinearLocator para controlar el número y formato de las marcas en el eje vertical.

**Pasos del ejercicio:**

1. **Generación de los datos**: Usaremos una función matemática para crear la superficie. La función es Z=sin⁡(X2+Y2)Z = \sin(\sqrt{X^2 + Y^2})Z=sin(X2+Y2​), lo que genera una superficie ondulada en 3D.
2. **Creación de la malla**: Usaremos numpy.meshgrid para crear una malla de puntos (X, Y) sobre la cual evaluaremos la función para obtener los valores de Z.
3. **Superficie 3D y mapa de colores**: Utilizaremos la función plot\_surface de Matplotlib para crear la superficie 3D y aplicaremos un mapa de colores (coolwarm), donde los valores bajos estarán en azul y los altos en rojo.
4. **Antialiased**: Configuraremos antialiased=False para que la superficie sea opaca, lo que mejora la visualización de la malla sin suavizar los bordes.
5. **Etiquetas personalizadas del eje Z**: Usaremos LinearLocator para controlar el número y formato de las marcas en el eje Z.
6. **Barra de colores**: Incluiremos una barra de colores que nos ayudará a interpretar los valores de Z en la superficie.

**GRAFICO**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Gráfico, Gráfico de superficie

Descripción generada automáticamente

# Parte 2

En la segunda parte una vez realizado las gráficas de los datos, se deberán guardar los archivos en el repositorio, deberás anexar la captura de pantalla con el nombre del archivo.

# Ejercicio de práctica 2.

Una vez realizado los escenarios, deberás guardar los archivos en el repositorio, anexar la captura de pantalla con el nombre del archivo ya guardado en el repositorio.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteRealización de la unidad del curso de AWS o Cisco y anexar captura del avance del curso.