Aldo Daniel Villaseñor Fierro

A01637907

In [ ]:
```
import numpy as np
import scipy.stats as stats
import plotly.graph_objects as go
import plotly
plotly.offline.init_notebook_mode(connected=True)
```

1. Hallar el procedimiento para el cálculo de probabilidad de que $P(X1 <= 2, X2 <= 3)$ con $X1$, $X2$ se distribuyen Normal con $\mu = (\mu_1 = 2.5, \mu_2 = 4)$ y

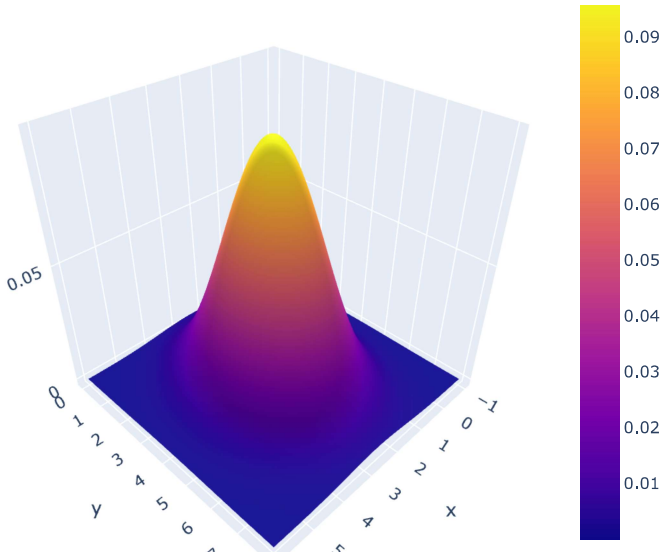
$$\Sigma = \begin{bmatrix} 1.2 & 0 \\ 0 & 2.3 \end{bmatrix} \tag{1}$$

In [ ]:
```
def pmnorm(x, miu, sigma):
    # Create a multivariate normal distribution object
    mvn = stats.multivariate_normal(mean=miu, cov=sigma)

    # Calculate the cumulative distribution function (CDF) for each quantile
    cdf = mvn.cdf(x)

    return cdf

# Example usage
x = np.array([2,3])  # Quantiles
miu = np.array([2.5,4])  # Mean vector
sigma = np.array([[1.2, 0],[0,2.3]])  # Covariance matrix

# Create a multivariate normal distribution object
mvn = stats.multivariate_normal(mean=miu, cov=sigma)

# Calculate the cumulative distribution function (CDF) for each quantile
cdf = mvn.cdf(x)

print(f'Probabilidad de que x1<=2 y x2<=3{cdf}')
```

```
Probabilidad de que x1<=2 y x2<=30.08257333341548989
```

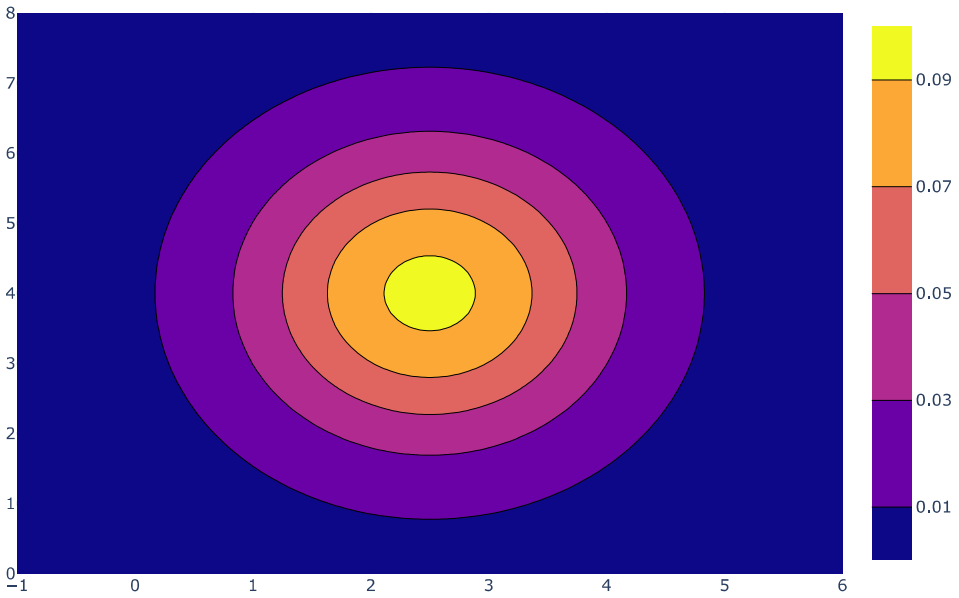
2. Grafique la anterior distribución bivariada del problema 1

In [ ]:
```
x1=np.linspace(-1,6,100)
x2=np.linspace(0,8,100)
# 3D plot of the multivariate normal distribution function with plotly.express
X1, X2 = np.meshgrid(x1, x2)
Z = mvn.pdf(np.dstack((X1, X2)))
#ig = px.surface(x=X1, y=X2, z=Z, title='Multivariate Normal Distribution')
fig = go.Figure(data=[go.Surface(z=Z, x=X1, y=X2)],layout=go.Layout(width=600, height=600))
fig.show()
```



3. Grafique los contornos de la anterior distribución normal bivariada correspondiente a las alturas de 0.01, 0.03, 0.05, 0.07, 0.09

In [ ]:
```
fig = go.Figure(data=[go.Contour(z=Z,x=x1,y=x2,
                                 contours=dict(start=0.01, end=0.09, size=0.02))],layout=go.Layout(width=800, height=600))
fig.show()
```



In [ ]: