Laboratorio 6: MODELO MATEMÁTICO. PROGRAMACIÓN LINEAL. VARIABLES DE DECISIÓN. FUNCIÓN OBJETIVO. RESTRICCIONES. CLASIFICACIÓN DE LOS MODELOS

Integrantes:

- Aranda Huerta, Milene
- Escriba Flores, Daniel Agustin

In [1]:

```python
## Importamos las librerias necesarias
from IPython.display import Image
import pulp
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import linprog

# ================================
import warnings
warnings.filterwarnings('ignore')
```

PROBLEMATICA

In [2]:

```python
Image("problema.png")
```

Out[2]:

Snorlax S. A. produce pinturas para interiores y exteriores con dos materias primas, M1 y M2. La tabla mostrada proporciona información básica del problema:

Materia prima	Toneladas de materia prima por tonelada de:		Disponibilidad diaria máxima (toneladas)
	Pintura para exteriores	Pintura para interiores	
M1	6	4	24
M2	1	2	6
Utilidad por tonelada (miles de dólares)	5	4	

Por estudios previos se estimó que la diferencia entre la demanda de pinturas para interiores y exteriores debe ser menor o igual a una tonelada. Además, la demanda diaria máxima de pinturas para interiores es de 2 toneladas.
De acuerdo a lo anterior, haga lo siguiente:

Parte A

Escriba el problema de programación lineal y péguelo como imagen

In [3]:
```
Image("prograline.png")
```

Out[3]:

Variables de decisión:

x_1: Toneladas de pintura para exteriores a producir por día

x_2: Toneladas de pintura para interiores a producir por día

Función Objetivo:

$$Z = 5x_1 + 4x_2$$

Restricciones:

$$6x_1 + 4x_2 \leq 24$$

$$x_1 + 2x_2 \leq 6$$

$$x_2 - x_1 \leq 1$$

$$x_2 \leq 2$$

No negatividad:

$$x_1 \geq 0 \,;\, x_2 \geq 0$$

PARTE B

Utilizando la librería ‘PuLP’, encuentre los valores que maximicen las utilidades, así como la utilidad máxima

In [4]:
```python
problema = pulp.LpProblem("utilidades",pulp.LpMaximize)

# Variables de decisión y marcamos la no negatividad

x1 = pulp.LpVariable("x1",lowBound = 0)
x2 = pulp.LpVariable("x2",lowBound = 0)

# Función objetivo
problema += 5*x1 + 4*x2

#Restricciones
problema += 6*x1 +  4*x2 <= 24
problema +=   x1 +  2*x2 <=  6
problema +=   x2 -    x1 <=  1
problema +=           x2 <=  2

# Resolvemos y mostramos la cantidad de soluciones
sol = problema.solve()
sol
```

Out[4]:
```
1
```

In [5]:
```python
print('z_max = {0:.4f}, x1 = {1:.4f}, x2 = {2:.4f}'.
     format(pulp.value(problema.objective),pulp.value(x1),pulp.value(x2)))
```

```
z_max = 21.0000, x1 = 3.0000, x2 = 1.5000
```

Los resultados satisfacen las restricciones del problema. La diferencia x2 - x1 = -1.5 muestra que x2 será siempre como máximo 1 tonelada mayor que x1.

PARTE C

Utilizando la librería 'SciPy', encuentre los valores que maximicen las utilidades, así como la utilidad máxima

In [6]:
```python
z = np.array([-5,-4]) # coeficientes de la funcion objetivo
A = np.array([[6,4],[1,2],[-1,1],[0,1]]) # coeficientes de las restriciones
b = np.array([24, 6, 1, 2]) #Terminos independientes de las restricciones

resul = linprog(z,A,b, method="revised simplex")
resul
```

Out[6]:
```
 message: Optimization terminated successfully.
 success: True
  status: 0
     fun: -21.0
       x: [ 3.000e+00  1.500e+00]
     nit: 2
```

In [7]:
```python
print('z_max = {0:.4f}, x1 = {1:.4f}, x2 = {2:.4f}'.
     format(-resul.fun,resul.x[0],resul.x[1]))
```

```
z_max = 21.0000, x1 = 3.0000, x2 = 1.5000
```

Conclusión

El problema de programación lineal planteado se resolvió exitosamente utilizando las librerías PuLP y SciPy, obteniendo resultados coincidentes. La solución óptima encontrada satisface las restricciones establecidas. Además, la coincidencia de resultados valida la precisión y robustez de la solución óptima obtenida