

Universidad Nacional del Altiplano  
Facultad de Ingeniería Estadística e Informática

**Docente:** Fred Torres Cruz

**Estudiante:** Luis Angel Quenaya Loza

**Código:** 241411

## Actividad N°03

### Gráficas de funciones lineales en Python

#### Problema 01

Un desarrollador dispone de un máximo de 15 horas semanales para dedicar al desarrollo de *front-end* ( $x$ ) y *back-end* ( $y$ ). Además, se tienen las siguientes condiciones:

- Debe dedicar al menos 5 horas al desarrollo de front-end:

$$x \geq 5$$

- El tiempo total invertido no puede superar las 15 horas:

$$x + y \leq 15$$

- Las variables no pueden ser negativas:

$$x \geq 0, \quad y \geq 0$$

Por lo tanto, el sistema de restricciones queda representado como:

$$\begin{cases} x \geq 5 \\ x + y \leq 15 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

#### Código en Python

```
1 def luis_expresion(expr: str) -> str:
2     """Convierte la expresi n a una forma v lida en Python."""
3     expr = expr.replace(" ", "")
4     expr = expr.replace("^", "**")
```

```
5     expr = expr.replace("-x", "-1*x")
6     expr = expr.replace("+x", "+1*x")
7
8     if expr.startswith("x"):
9         expr = "1*" + expr
10
11     expr = expr.replace("x", "*x")
12     expr = expr.replace("**x", "*x")
13     return expr
14
15
16 def evaluar(expr, x):
17     """Eval a la funci n en un valor de x."""
18     try:
19         return eval(expr)
20     except:
21         return None
22
23
24 def graficar_ascii(func1, xmin=0, xmax=20, ymin=0, ymax=20):
25     """Dibuja una funci n lineal en consola usando ASCII, mostrando
26         la regi n factible."""
27     for y in range(ymax, ymin - 1, -1):
28         linea = ""
29         for x in range(xmin, xmax + 1):
30             y1 = evaluar(func1, x)
31
32             cond1 = (y1 is not None and round(y1) == y)
33             cond2 = (x == 5)
34             cond3 = (x >= 5 and y <= -x + 15 and y >= 0)
35
36             if cond1:
37                 linea += "1"
38             elif cond2:
39                 linea += "|"
40             elif cond3:
41                 linea += "*"
42             elif x == 0 and y == 0:
43                 linea += "+"
44             elif x == 0:
45                 linea += ":"
46             elif y == 0:
47                 linea += "-"
48             else:
49                 linea += " "
50     print(linea)
```

```

51     print("\nLeyenda del gráfico:")
52     print(" 1 = Recta y = -x + 15 (tiempo total)")
53     print(" | = Restricción x = 5 (mínimo front-end)")
54     print(" * = Zona factible (combinaciones posibles)")
55     print(" + = Origen (0,0)")
56
57
58
59 print("Modelo de horas del desarrollador")
60 f1 = "-x+15"
61
62 f1 = luis_expresion(f1)
63
64 graficar_ascii(f1)

```

```

Modelo de horas del desarrollador
: |
: |
: |
: |
: |
: |
1 |
:1 |
: 1 |
: 1 |
: 1 |
: 1 |
: |1
: |*1
: |**1
: |***1
: |****1
: |*****1
: |*****1
: |*****1
: |*****1
: |*****1
+---|*****1-----

Leyenda del gráfico:
1 = Recta y = -x + 15 (tiempo total)
| = Restricción x = 5 (mínimo front-end)
* = Zona factible (combinaciones posibles)
+ = Origen (0,0)
>>>|

```

Figura 1: Gráfico generado por el programa en Python

## Interpretación del gráfico

El gráfico muestra la región factible de asignación de horas entre el desarrollo de **front-end** ( $x$ ) y **back-end** ( $y$ ).

Las restricciones son:

$$x \geq 5, \quad x + y \leq 15, \quad x, y \geq 0$$

La recta  $y = -x + 15$  indica el límite del tiempo total, mientras que la línea  $x = 5$  asegura el mínimo de horas en front-end. La zona válida es el triángulo formado entre ambas rectas en el primer cuadrante.

En total, existen **66 combinaciones posibles** de horas que cumplen con estas condiciones.

## Problema 02

Un ingeniero de datos administra dos tipos de servidores en la nube: **Servidores A** y **Servidores B**. El costo por hora de Servidor A es de S/3 y el de Servidor B es de S/5. El presupuesto máximo semanal asignado para mantener los servidores es de S/20.

Se requiere determinar cuántas horas puede mantener activos cada tipo de servidor, formular el sistema de restricciones correspondiente y representarlo gráficamente.

### Restricciones

Sea  $x$  el número de horas de uso de los **Servidores A** y  $y$  el número de horas de uso de los **Servidores B**.

Las restricciones son:

$$\begin{aligned}3x + 5y &\leq 20 \\ x &\geq 0 \\ y &\geq 0\end{aligned}$$

## Código en Python

```
1 def luis_expresion(expr: str) -> str:
2     """Convierte la expresi n en formato Python v lido."""
3     expr = expr.replace(" ", "")
4     expr = expr.replace("^", "**")
5     expr = expr.replace("-x", "-1*x")
6     expr = expr.replace("+x", "+1*x")
7     if expr.startswith("x"):
8         expr = "1*" + expr
9     expr = expr.replace("x", "*x")
10    expr = expr.replace("**x", "*x")
11    return expr
12
13 def evaluar(expr, x):
14     """Eval a la funci n en un valor de x."""
15     try:
16         return eval(expr)
17     except:
18         return None
19
20 def graficar_ascii(func1, func2, xmin=0, xmax=20, ymin=0, ymax=10):
21     """Dibuja dos funciones lineales en ASCII con n meros en los
    ejes."""
```

```

22     for y in range(ymax, ymin - 1, -1):
23         linea = f"{y:2} "
24         for x in range(xmin, xmax + 1):
25             y1 = evaluar(func1, x)
26             y2 = evaluar(func2, x)
27
28             cond1 = (y1 is not None and round(y1) == y)
29             cond2 = (y2 is not None and round(y2) == y)
30
31             if cond1 and cond2:
32                 linea += "X"
33             elif cond1:
34                 linea += "1"
35             elif cond2:
36                 linea += "2"
37             elif x == 0 and y == 0:
38                 linea += "+"
39             elif x == 0:
40                 linea += "|"
41             elif y == 0:
42                 linea += "-"
43             else:
44                 linea += " "
45         print(linea)
46
47
48     eje_x = "    "
49     for x in range(xmin, xmax + 1):
50         if x % 2 == 0:
51             eje_x += str(x % 10)
52         else:
53             eje_x += " "
54     print(eje_x)
55
56     print("\nLeyenda del gr fico:")
57     print(" 1 = Restricci n 3x + 5y = 20")
58     print(" 2 = Eje base (y=0)")
59     print(" X = Intersecci n")
60     print(" | = Eje Y")
61     print(" - = Eje X")
62     print(" + = Origen (0,0)")
63
64
65     print("Modelo de servidores en la nube")
66     f1 = "-(3/5)*x + 4"
67     f2 = "0"
68

```

```

69 f1 = luis_expresion(f1)
70 f2 = luis_expresion(f2)
71
72 graficar_ascii(f1, f2)

```

Modelo de servidores en la nube

```

10 |
  9 |
  8 |
  7 |
  6 |
  5 |
  4 1
  3 |11
  2 |  11
  1 |    1
  0 222222XX222222222222
    0 2 4 6 8 0 2 4 6 8 0

```

Leyenda del gráfico:

```

1 = Restricción 3x + 5y = 20
2 = Eje base (y=0)
X = Intersección
| = Eje Y
- = Eje X
+ = Origen (0,0)

```

Figura 2: Gráfico generado por el programa en Python

### Interpretación del gráfico

El gráfico obtenido representa las combinaciones posibles de horas de uso entre los **Servidores A** ( $x$ ) y los **Servidores B** ( $y$ ), teniendo en cuenta el presupuesto máximo semanal de S/20.

La recta:

$$3x + 5y = 20 \quad \Longleftrightarrow \quad y = -\frac{3}{5}x + 4$$

marca el límite del gasto.

- Si se usan solo **Servidores A**, el máximo posible es

$$x = \frac{20}{3} \approx 6,6 \text{ horas.}$$

- Si se usan solo **Servidores B**, el máximo posible es

$$y = 4 \text{ horas.}$$

En la zona bajo la recta y dentro del primer cuadrante se encuentran todas las combinaciones válidas. Por ejemplo:

$$(x = 5, y = 1) \quad \text{o} \quad (x = 2, y = 2,8)$$

son posibles porque no superan el presupuesto.

## Problema 03

Un administrador de proyectos tecnológicos organiza su tiempo entre reuniones con stakeholders ( $x$ ) y trabajo en la documentación técnica ( $y$ ). Las reuniones requieren al menos 4 horas semanales y la documentación al menos 6 horas. Si dispone de 12 horas para ambas actividades, determine la región factible y analice las combinaciones posibles de tiempo.

### Restricciones

Sean  $x$  las horas destinadas a reuniones y  $y$  las horas para documentación técnica.

$$\begin{cases} x + y \leq 12 \\ x \geq 4 \\ y \geq 6 \\ x \geq 0, \quad y \geq 0 \end{cases}$$

### Código en Python

```

1 def luis_expresion(expr: str) -> str:
2     """Convierte la expresi n en formato v lido de Python."""
3     expr = expr.replace("^", "**")
4     expr = expr.replace("-x", "-1*x")
5     expr = expr.replace("+x", "+1*x")
6     if expr.startswith("x"):
7         expr = "1*" + expr
8     expr = expr.replace("x", "*x")
9     expr = expr.replace("**x", "*x")
10    return expr
11
12 def evaluar(expr, x):
13     """Eval a la funci n en un valor de x."""
14     try:
15         return eval(expr)
16     except:
17         return None
18
19 def graficar_ascii(func1, func2, xmin=0, xmax=12, ymin=0, ymax=12):
20     """Dibuja restricciones en ASCII con n meros en los ejes."""

```

```
21     for y in range(ymax, ymin - 1, -1):
22         linea = f"{y:2} "
23         for x in range(xmin, xmax + 1):
24             y1 = evaluar(func1, x)
25             y2 = evaluar(func2, x)
26
27             cond1 = (y1 is not None and round(y1) == y)
28             cond2 = (y2 is not None and round(y2) == y)
29
30             if cond1:
31                 linea += "1"
32             elif cond2:
33                 linea += "2"
34             elif x == 0 and y == 0:
35                 linea += "+"
36             elif x == 0:
37                 linea += "|"
38             elif y == 0:
39                 linea += "-"
40             else:
41                 linea += " "
42         print(linea)
43
44     eje_x = "    "
45     for x in range(xmin, xmax + 1):
46         eje_x += str(x % 10)
47     print(eje_x)
48
49     print("\nLeyenda del gráfico:")
50     print("1 = Restricción x + y = 12")
51     print("2 = Restricciones mínimas (x=4, y=6)")
52     print("| = Eje Y, - = Eje X, + = Origen (0,0)")
53
54     print("Modelo de tiempo del administrador")
55     f1 = "-x + 12"
56     f2 = "0"
57
58     f1 = luis_expresion(f1)
59     f2 = luis_expresion(f2)
60
61     graficar_ascii(f1, f2)
```



## Figura

```

Modelo de tiempo del administrador
12 1
11 |1
10 | 1
 9 | 1
 8 | 1
 7 | 1
 6 | 1
 5 | 1
 4 | 1
 3 | 1
 2 | 1
 1 | 1
 0 222222222221
    0123456789012

Leyenda del gráfico:
1 = Restricción  $x + y = 12$ 
2 = Restricciones mínimas ( $x=4$ ,  $y=6$ )
| = Eje Y, - = Eje X, + = Origen (0,0)
>>

```

Figura 3: Enter Caption

Figura 4: Región factible del Problema 03

## Interpretación del gráfico

El gráfico muestra la región de combinaciones posibles de horas para reuniones y documentación. Si solo se realizan reuniones, como mínimo deben ser  $x = 4$  horas; si solo se trabaja en documentación, se requieren al menos  $y = 6$  horas. La recta  $x + y = 12$  marca el límite del tiempo disponible. La zona factible es el área comprendida por estas restricciones, donde se hallan todas las distribuciones válidas de tiempo.

## Problema 04

Una empresa de desarrollo de videojuegos produce dos tipos de *assets*: Modelos 3D ( $x$ ) y Texturas ( $y$ ). Cada modelo 3D requiere 2 horas de trabajo y cada textura requiere 3 horas. El equipo de arte dispone de 18 horas semanales. Formule las restricciones, represéntelas gráficamente y determine cuántos *assets* de cada tipo pueden producirse.

## Restricciones

Sean  $x$  el número de modelos 3D y  $y$  el número de texturas.

$$\begin{cases} 2x + 3y \leq 18 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

## Código en Python

```
1 def luis_expression(expr: str) -> str:
2     expr = expr.replace("^", "**")
3     expr = expr.replace("-x", "-1*x")
4     expr = expr.replace("+x", "+1*x")
5     if expr.startswith("x"):
6         expr = "1*" + expr
7     expr = expr.replace("x", "*x")
8     expr = expr.replace("**x", "*x")
9     return expr
10
11 def evaluar(expr, x):
12     try:
13         return eval(expr)
14     except:
15         return None
16
17 def graficar_ascii(func1, func2, xmin=0, xmax=10, ymin=0, ymax=10):
18     for y in range(ymax, ymin - 1, -1):
19         linea = f"{y:2} "
20         for x in range(xmin, xmax + 1):
21             y1 = evaluar(func1, x)
22             y2 = evaluar(func2, x)
23
24             cond1 = (y1 is not None and round(y1) == y)
25             cond2 = (y2 is not None and round(y2) == y)
26
27             if cond1:
28                 linea += "1"
29             elif cond2:
30                 linea += "2"
31             elif x == 0 and y == 0:
32                 linea += "+"
33             elif x == 0:
34                 linea += "|"
35             elif y == 0:
36                 linea += "-"
37             else:
38                 linea += " "
39         print(linea)
40
```

```

41     eje_x = "      "
42     for x in range(xmin, xmax + 1):
43         eje_x += str(x % 10)
44     print(eje_x)
45
46     print("\nLeyenda del gráfico:")
47     print("1 = Restricción  $2x + 3y = 18$ ")
48     print("2 = Eje base ( $y=0$ )")
49     print("| = Eje Y, - = Eje X, + = Origen (0,0)")
50
51 print("Modelo de producción de assets")
52 f1 = " $-(2/3)x + 6$ "
53 f2 = "0"
54
55 f1 = luis_expression(f1)
56 f2 = luis_expression(f2)
57
58 graficar_ascii(f1, f2)

```

## Figura

```

Modelo de producción de assets
10 |
  9 |
  8 |
  7 |
  6 1
  5 |11
  4 |  1
  3 |   11
  2 |    1
  1 |     11
  0 22222222212
    01234567890

Leyenda del gráfico:
1 = Restricción  $2x + 3y = 18$ 
2 = Eje base ( $y=0$ )
| = Eje Y, - = Eje X, + = Origen (0,0)
|

```

Figura 5: Región factible del Problema 04

## Interpretación del gráfico

El gráfico representa todas las combinaciones de producción de modelos 3D y texturas que respetan las 18 horas disponibles. Si solo se producen modelos 3D, el máximo posible es  $x = 9$ ; si solo se producen texturas, el máximo es  $y = 6$ . La región bajo la recta es el conjunto de combinaciones viables de producción.

## Problema 05

Una startup de hardware dispone de un máximo de 50 unidades de componentes electrónicos. Para ensamblar un dispositivo tipo A ( $x$ ) se necesitan 5 unidades, mientras que para un dispositivo tipo B ( $y$ ) se requieren 10 unidades. Determine cuántos dispositivos de cada tipo pueden ensamblar sin exceder las 50 unidades, represéntelo gráficamente y explique las combinaciones de producción.

### Restricciones

Sean  $x$  el número de dispositivos tipo A y  $y$  el número de dispositivos tipo B.

$$\begin{cases} 5x + 10y \leq 50 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

### Código en Python

```
1 def luis_expresion(expr: str) -> str:
2     expr = expr.replace("^", "**")
3     expr = expr.replace("-x", "-1*x")
4     expr = expr.replace("+x", "+1*x")
5     if expr.startswith("x"):
6         expr = "1*" + expr
7     expr = expr.replace("x", "*x")
8     expr = expr.replace("**x", "*x")
9     return expr
10
11 def evaluar(expr, x):
12     try:
13         return eval(expr)
14     except:
15         return None
16
17 def graficar_ascii(func1, func2, xmin=0, xmax=12, ymin=0, ymax=6):
18     for y in range(ymax, ymin - 1, -1):
19         linea = f"{y:2} "
20         for x in range(xmin, xmax + 1):
```

```
21     y1 = evaluar(func1, x)
22     y2 = evaluar(func2, x)
23
24     cond1 = (y1 is not None and round(y1) == y)
25     cond2 = (y2 is not None and round(y2) == y)
26
27     if cond1:
28         linea += "1"
29     elif cond2:
30         linea += "2"
31     elif x == 0 and y == 0:
32         linea += "+"
33     elif x == 0:
34         linea += "|"
35     elif y == 0:
36         linea += "-"
37     else:
38         linea += " "
39     print(linea)
40
41     eje_x = "    "
42     for x in range(xmin, xmax + 1):
43         eje_x += str(x % 10)
44     print(eje_x)
45
46     print("\nLeyenda del gráfico:")
47     print("1 = Restricción  $5x + 10y = 50$ ")
48     print("2 = Eje base ( $y=0$ )")
49     print("| = Eje Y, - = Eje X, + = Origen (0,0)")
50
51     print("Modelo de ensamblaje de dispositivos")
52     f1 = " $-(1/2)x + 5$ "
53     f2 = "0"
54
55     f1 = luis_expression(f1)
56     f2 = luis_expression(f2)
57
58     graficar_ascii(f1, f2)
```

## Figura

```

Modelo de ensamblaje de dispositivos
 6 |
 5 | 1
 4 |111
 3 |   1
 2 |   111
 1 |       1
 0 2222222221112
    0123456789012

Leyenda del gráfico:
1 = Restricción  $5x + 10y = 50$ 
2 = Eje base ( $y=0$ )
| = Eje Y, - = Eje X, + = Origen (0,0)

```

Figura 6: Región factible del Problema 05

## Interpretación del gráfico

El gráfico muestra las combinaciones posibles de dispositivos A y B dentro del límite de 50 componentes. Si solo se ensamblan dispositivos A, el máximo es  $x = 10$ ; si solo se ensamblan dispositivos B, el máximo es  $y = 5$ . La zona bajo la recta representa todas las combinaciones factibles de producción.