

# Restricciones y Sistemas de Ecuaciones

Ccoarite Dueñas Henry

## I. Resolución de Ejercicios

Los problemas requieren la formulación de sistemas de **restricciones de desigualdad** para definir la **región factible**.

### Ejercicio 1: Desarrollo de Software

**Enunciado:** Un desarrollador tiene un máximo de **15 horas semanales** para dedicar al desarrollo de **front-end** ( $x$ ) y **back-end** ( $y$ ). Debe dedicar **al menos 5 horas** al front-end. Formule el sistema de restricciones y represente la región factible.

1. **Definición de Variables:**  $x$ : Horas Front-end,  $y$ : Horas Back-end.

2. **Restricciones:**

- $x + y \leq 15$
- $x \geq 5$
- $y \geq 0$

3. **Sistema:**

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

4. **Región Factible:** Vértices  $(5, 0)$ ,  $(15, 0)$ ,  $(5, 10)$ .

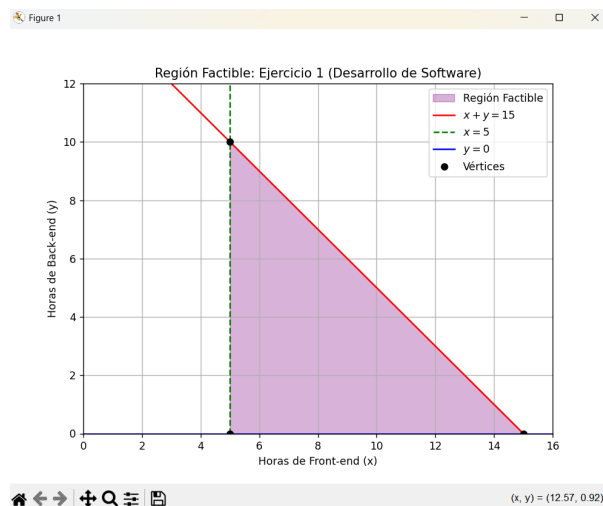


Figure 1: Región factible del Ejercicio 1: Desarrollo de Software.

## Ejercicio 2: Administración de Servidores en la Nube

**Enunciado:** El costo por hora de **Servidor A** ( $x$ ) es **\$3** y de **Servidor B** ( $y$ ) es **\$5**. El presupuesto máximo semanal es de **\$20**.

### 1. Restricciones:

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

### 2. Intersecciones:

- Eje  $x$ :  $\frac{20}{3} \approx 6.67$
- Eje  $y$ : 4

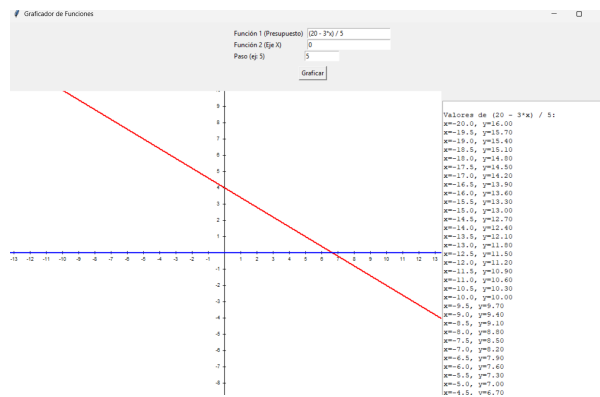


Figure 2: Región factible del Ejercicio 2: Administración de Servidores en la Nube.

## Ejercicio 3: Administración de Proyectos

**Enunciado:** Un administrador organiza su tiempo entre reuniones ( $x$ ) y documentación técnica ( $y$ ). Requiere al menos 4 y 6 horas respectivamente, y dispone de 12 horas en total.

### 1. Restricciones:

- $x \geq 4$
- $y \geq 6$
- $x + y \leq 12$

### 2. Vértices: (4, 6), (4, 8), (6, 6).

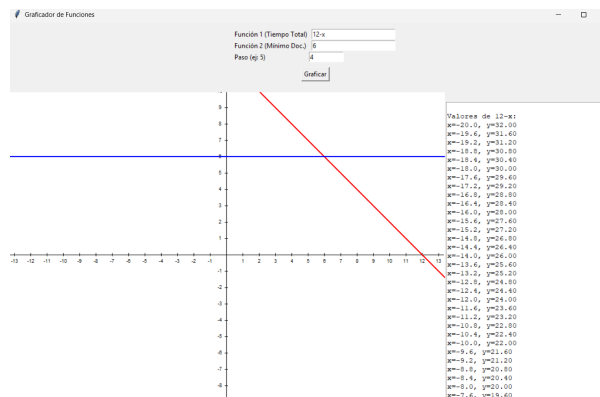


Figure 3: Región factible del Ejercicio 3: Administración de Proyectos.

## Ejercicio 4: Producción de Assets para Videojuegos

**Enunciado:** Una empresa produce Modelos 3D ( $x$ ) y Texturas ( $y$ ). Cada modelo requiere 2 horas, cada textura 3 horas y hay 18 horas disponibles.

1. Restricciones:

$$\begin{aligned} 2x + 3y &\leq 18 \\ x, y &\geq 0 \end{aligned}$$

2. Puntos factibles: (9, 0), (6, 2), (3, 4), (0, 6).

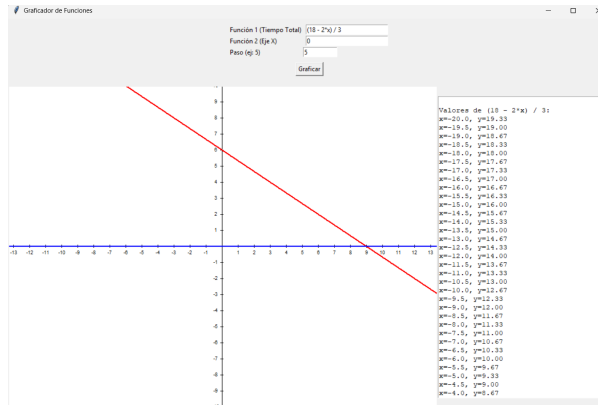


Figure 4: Región factible del Ejercicio 4: Producción de Assets para Videojuegos.

## II. Códigos de Apoyo para Graficación

### A. Graficador de Igualdades (tkinter)

```
1 import tkinter as tk
2 import math
3 from tkinter import messagebox
4
5 def evaluar(expr: str, x: float):
6     try:
7         return eval(expr, {"x": x, "math": math, "__builtins__": {}})
8     except Exception:
9         return None
10
11 def dibujar_ejes(canvas, ancho, alto, escala):
12     cx, cy = ancho // 2, alto // 2
13     canvas.create_line(0, cy, ancho, cy, fill="black")
14     canvas.create_line(cx, 0, cx, alto, fill="black")
15
16 def trazar_funcion(canvas, expr, color, ancho, alto, escala, paso):
17     cx, cy = ancho // 2, alto // 2
18     puntos = []
19     for x in range(-200, 201, paso):
20         x_real = x / 10
21         y_real = evaluar(expr, x_real)
22         if y_real is not None:
23             px = cx + x_real * escala
24             py = cy - y_real * escala
25             puntos.append((px, py))
26     if len(puntos) > 1:
27         canvas.create_line(puntos, fill=color, width=2, smooth=True)
28
29 def graficar():
30     canvas.delete("all")
31     ancho, alto, escala, paso = 800, 600, 30, 5
```

```

32     dibujar_ejes(canvas, ancho, alto, escala)
33     trazar_funcion(canvas, entrada1.get(), "red", ancho, alto, escala, paso)
34     trazar_funcion(canvas, entrada2.get(), "blue", ancho, alto, escala, paso)
35
36 root = tk.Tk()
37 root.title("Graficador de Funciones")
38 panel = tk.Frame(root, padx=10, pady=10)
39 panel.pack()
40
41 tk.Label(panel, text="Funci n 1:").grid(row=0, column=0)
42 entrada1 = tk.Entry(panel, width=25)
43 entrada1.insert(0, "2*x+3")
44 entrada1.grid(row=0, column=1)
45
46 tk.Label(panel, text="Funci n 2:").grid(row=1, column=0)
47 entrada2 = tk.Entry(panel, width=25)
48 entrada2.insert(0, "-x+2")
49 entrada2.grid(row=1, column=1)
50
51 tk.Button(panel, text="Graficar", command=graficar).grid(row=2, column=0, columnspan=2)
52 canvas = tk.Canvas(root, width=800, height=600, bg="white")
53 canvas.pack()
54 root.mainloop()

```

Listing 1: Interfaz gráfica con Tkinter para trazar ecuaciones

## B. Graficación de Regiones Factibles (matplotlib)

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def graficar_region_factible_generico():
5     x = np.linspace(0, 16, 500)
6     y_superior = 15 - x
7     x_min = 5
8     y_inferior = 0 * x
9
10    x_f = x[x >= x_min]
11    y_sup = y_superior[x >= x_min]
12
13    plt.figure(figsize=(8, 6))
14    plt.fill_between(x_f, 0, y_sup, color='purple', alpha=0.3, label='Regi n factible')
15    plt.plot(x, y_superior, color='red', label='$x + y = 15$')
16    plt.axvline(x=x_min, color='green', linestyle='--', label='$x = 5$')
17    plt.axhline(0, color='blue')
18    plt.title('Regi n Factible (Ejemplo Gen rico)')
19    plt.xlabel('x')
20    plt.ylabel('y')
21    plt.legend()
22    plt.grid(True)
23    plt.show()
24
25 if __name__ == '__main__':
26     graficar_region_factible_generico()

```

Listing 2: Graficación de regiones factibles con Matplotlib