

## CONTROL DIFUSO

```
In [20]: 1 # CONTROL DIFUSO
2
3 # Encontrar valor de la propina a partir de la calidad del
4 # servicio y de la comida en un restaurante
5
6 # Importar Librerías
7 import numpy as np
8 import skfuzzy as fuzz
9 import matplotlib.pyplot as plt
10
11 # Generar variables del universo
12 # * Calidad y servicio en rangos subjetivos [0, 10]
13 # * La propina tiene un rango de [0, 25] en unidades de puntos porcentuales
14 x_calidad = np.arange(0, 11, 1)
15 x_servicio = np.arange(0, 11, 1)
16 x_propina = np.arange(0, 26, 1)
17
18 # Generar funciones de pertenencia difusas
19 calidad_baja = fuzz.trimf(x_calidad, [0, 0, 5])
20 calidad_media = fuzz.trimf(x_calidad, [0, 5, 10])
21 calidad_alta = fuzz.trimf(x_calidad, [5, 10, 10])
22 servicio_bajo = fuzz.trimf(x_servicio, [0, 0, 5])
23 servicio_medio = fuzz.trimf(x_servicio, [0, 5, 10])
24 servicio_alto = fuzz.trimf(x_servicio, [5, 10, 10])
25 propina_baja = fuzz.trimf(x_propina, [0, 0, 13])
26 propina_media = fuzz.trimf(x_propina, [0, 13, 25])
27 propina_alta = fuzz.trimf(x_propina, [13, 25, 25])
28
29 # Visualizar estos universos y funciones de pertenencia.
30 fig, (ax0, ax1, ax2) = plt.subplots(nrows=3, figsize=(8, 9))
31 ax0.plot(x_calidad, calidad_baja, 'b', linewidth=1.5, label='Mala')
32 ax0.plot(x_calidad, calidad_media, 'g', linewidth=1.5, label='Aceptable')
33 ax0.plot(x_calidad, calidad_alta, 'r', linewidth=1.5, label='Buena')
34 ax0.set_title('Calidad de la comida')
35 ax0.legend()
36
37 ax1.plot(x_servicio, servicio_bajo, 'b', linewidth=1.5, label='Malo')
38 ax1.plot(x_servicio, servicio_medio, 'g', linewidth=1.5, label='Aceptable')
39 ax1.plot(x_servicio, servicio_alto, 'r', linewidth=1.5, label='Excelente')
40 ax1.set_title('Calidad del servicio')
41 ax1.legend()
42
43 ax2.plot(x_propina, propina_baja, 'b', linewidth=1.5, label='Bajo')
44 ax2.plot(x_propina, propina_media, 'g', linewidth=1.5, label='Medio')
45 ax2.plot(x_propina, propina_alta, 'r', linewidth=1.5, label='Alto')
46 ax2.set_title('Valor de la propina')
47 ax2.legend()
48
49 # Ocultar los ejes superior / derecho
50 for ax in (ax0, ax1, ax2):
51     ax.spines['top'].set_visible(False)
52     ax.spines['right'].set_visible(False)
53     ax.get_xaxis().tick_bottom()
54     ax.get_yaxis().tick_left()
55 plt.tight_layout()
```