

In [2]:

```

1 from tkinter import messagebox, ttk
2 from tkinter import *
3 import tkinter
4 import matplotlib.pyplot as plt
5 import numpy as np
6 # http://pythondiario.com/2018/09/incrustar-graficos-matplotlib-en.html
7 #from tkinter.messagebox import showinfo
8 # http://www.boletin.upiita.ipn.mx/index.php/ciencia/845-cyt-numero-75/1760-como-diseno

```

In [3]:

```

1 #SE GENERA LA TABLA
2
3 tabla = (["Temperatura","Humedad","RPM Motor"],["Baja","Alta","Baja"],["Media","Alta","Alta"],
4          ["Media","Media","Baja"],["Alta","Media","Media"],["Baja","Baja","Baja"],
5          ["Media","Baja","Baja"],["Alta","Baja","Alta"])
6 for t in tabla:
7     print("|",t[0],"|",t[1],"|",t[2])
8

```

Temperatura	Humedad	RPM Motor
Baja	Alta	Baja
Media	Alta	Media
Alta	Alta	Media
Baja	Media	Baja
Media	Media	Baja
Alta	Media	Media
Baja	Baja	Baja
Media	Baja	Baja
Alta	Baja	Alta

In [5]:



```

1  #VENTANA PRINCIPAL
2  root = tkinter.Tk()
3  root.title("LOGICA DIFUSA")
4  root.geometry("800x800")
5  nb = ttk.Notebook(root)
6  nb.pack(fill='both', expand='yes')
7  p1 = ttk.Frame(nb)
8  p2 = ttk.Frame(nb)
9  p3 = ttk.Frame(nb)
10 nb.add(p1, text='Temperatura')
11 nb.add(p2, text='Humedad')
12 nb.add(p3, text='RPM Motor')
13 # campo_de_texto = ttk.Entry(p1)
14 # campo_de_texto2 = ttk.Entry(p2)
15 # campo_de_texto.pack()
16 # campo_de_texto2.pack()
17 texta = tkinter.Text(p1,height=2, width=30)
18 texta.pack(side='right',fill='both', expand='yes')
19 global a
20 a = 0
21 b = 0
22 mylist = Listbox(p1)
23 cont= 0
24 for line in range(100):
25     mylist.insert(cont,str(line))
26     cont=cont+1
27 mylist.pack( side = LEFT)
28 myl = Listbox(p1)
29 cont= 0
30 for line in range(10,100,5):
31     myl.insert(cont,str(line))
32     cont=cont+1
33 myl.pack( side = RIGHT)
34
35 valores= ([])
36 valoresh= ([])
37 def grafico_temp():
38     print("GRAFICA TEMPERATURA")
39     listax = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20, 10,11,12,13,14,15,
40     listay = [1,1,1,1,1,1,1,1,1,1,1,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0, 0,0.1,0.2,0
41     plt.plot(listax[:21], listay[:21], lw=5,label="Baja")
42     plt.plot(listax[21:62], listay[21:62], lw=5, label="Media")
43     plt.plot(listax[62:], listay[62:], lw=5, label="Alta")
44     plt.axis([min(listax),max(listax),min(listay),max(listay)])
45     plt.grid()
46     plt.legend(loc="lower right", title="Legend Title", frameon=False)
47     global a
48     a = int(mylist.get(int(mylist.curselection()[0])))
49     #SE BUSCA EL VALOR INGRESADO EN EL VECTOR DE X ESTA EN T BAJA
50     cont=0
51     for lx in listax[:21]:
52         if int(lx) == int(a):
53             print("el valor ingresado es BAJA>> ",lx,listay[cont])
54             valores.insert(0,["Baja",listay[cont]])
55             cont =cont +1
56     #SE BUSCA EL VALOR INGRESADO EN EL VECTOR DE X ESTA EN T MEDIA
57     cont1=0
58     lisyn = listay[21:62]
59     for lx in listax[21:62]:

```

```

60         if int(lx) == int(a):
61             print("el valor ingresado es MEDIA>> ",lx,lisyn[cont1])
62             valores.insert(1,["Media",lisyn[cont1]])
63             cont1 = cont1 +1
64         #SE BUSCA EL VALOR INGRESADO EN EL VECTOR DE X ESTA T ALTA
65         cont=0
66         lista = listay[62:]
67         for lx in listax[62:]:
68             if int(lx) == int(a):
69                 print("el valor ingresado es ALTO >> ",lx,lista[cont])
70                 valores.insert(2,["Alta",lista[cont]])
71                 cont =cont +1
72         plt.axvline(int(a), label='pyplot vertical line',color='red')
73         plt.show()
74         texta.insert(END, plt)
75         print(valores)
76
77 boton_tem = Button(p1, text="TEMPERATURA", command=grafico_temp)
78 boton_tem.pack()
79
80 def grafico_hume():
81     print(" GRAFICA HUMEDAD")
82     listax = [70, 60, 50,60, 50, 30, 20, 30,20,10]
83     listay = [1, 1, 0, 0, 1, 1, 0, 0, 1, 1]
84     plt.plot(listax[:3], listay[:3], lw=5,label="Alta")
85     plt.plot(listax[3:7], listay[3:7], lw=5, label="Media")
86     plt.plot(listax[7:], listay[7:], lw=5, label="Baja")
87     plt.axis([max(listax),min(listax),min(listay),max(listay)])
88     plt.grid(True)
89     plt.legend(loc="lower right", title="Legend Title", frameon=False)
90     b = int(my1.get(int(my1.curselection()[0])))
91     plt.axvline(int(b), label='pyplot vertical line',color='red')
92     #SE BUSCA EL VALOR INGRESADO EN EL VECTOR DE X ESTA T ALTA
93     cont=0
94     lista = listay[:3]
95     for lx in listax[:3]:
96         if int(lx) == int(b):
97             print("el valor ingresado es ALTO >> ",lx,lista[cont])
98             valoresh.insert(0,["Alta",lista[cont]])
99
100         cont =cont +1
101     #SE BUSCA EL VALOR INGRESADO EN EL VECTOR DE X ESTA EN T MEDIA
102     cont1=0
103     lisyn = listay[3:7]
104     for lx in listax[3:7]:
105         if int(lx) == int(b):
106             print("el valor ingresado es MEDIO>> ",lx,lisyn[cont1])
107             valoresh.insert(1,["Media",lisyn[cont1]])
108
109         cont1 = cont1 +1
110
111     #SE BUSCA EL VALOR INGRESADO EN EL VECTOR DE X ESTA EN T BAJA
112     cont=0
113     for lx in listax[7:]:
114         if int(lx) == int(b):
115             print("el valor ingresado es BAJO>> ",lx,listay[cont])
116             valoresh.insert(2,["Baja",listay[cont]])
117
118         cont =cont +1
119
120     #SE COMPRUEBA CUALES SON LOS RESULTADOS IGUALES EN LA TABLA

```

```

121 cv = 0
122 valores
123 valoresh
124 resultados = ([])
125 for vl in tabla:
126     #REGLA 1 SI LOS VALORES DE PUNTOS DE INTERCICION CON LAS RECTAS
127     #YA SEA MEDIA O BAJA DE LA TEMPERATURA Y HUMEDAD ESTA EN LA PRIMERA
128     #POSICION DE LOS VECTORES "valores" y "valoresh", COINCIDEN CON LOS
129     #VALORES DE CADA FILA "vl" DE LA TABLA "tabla"
130     if (vl[0] == valores[0][0] and vl[1] == valoresh[0][0]) or (vl[0] == valores[0
131         resultados.insert(cv,vl)
132         cv = cv +1
133     #REGLA 2 SI LOS VALORES que estan en la segunda posicion"1" DE PUNTOS DE INTE
134     #YA SEA MEDIA O BAJA DE LA TEMPERATURA Y HUMEDAD ESTA EN LA PRIMERA
135     #POSICION DE LOS VECTORES "valores" y "valoresh", COINCIDEN CON LOS
136     #VALORES DE CADA FILA "vl" DE LA TABLA "tabla"
137     elif (vl[0] == valores[1][0] and vl[1] == valoresh[0][0]) or (vl[0] == valores
138         resultados.insert(cv,vl)
139         cv = cv +1
140 for vl1 in resultados:
141     print("**** SE IMPRIME LOS VALORES DE IGUALDA >> ",vl1[0],vl1[1],vl1[2])
142 contf = 0
143 print("*****")
144 print(" Valores de TEMPERATURA",valores)
145 print("Valores de HUMEDAD",valoresh)
146 print("*****")
147 print("TABLA DE INFERENCIA ")
148 resul = []
149 for vf in valores:
150     print("X = ",vf[1],"Y = ",valoresh[contf][1])
151     #SE REALIZA LAS AGREGACIONES
152     agre = max(min(vf[1],valoresh[contf][1]),min(vf[1],valoresh[1][1]))
153     resul.append(agre)
154     contf =contf +1
155 print("*****")
156 print("SE ESCOJE EL MAXIMO DE LOS MINIMOS, HACIENDO LAS AGREGACIONES")
157 print(resul)
158 #SE REALIZA EL COLOREO DEL AREA
159 # plt.fill_between(2,8,color="green")
160 plt.show()
161
162 #SE GRAFICA LA TABLA DE RPM DEL MOTOR
163 print(" GRAFICA RPM DEL MOTOR")
164 listax = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20, 10,11,12,13,14,15,
165 listay = [1,1,1,1,1,1,1,1,1,1,1,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0, 0,0.1,0.2,0
166 plt.plot(listax[:21], listay[:21], lw=5,label="Baja")
167 plt.plot(listax[21:62], listay[21:62], lw=5, label="Media")
168 plt.plot(listax[62:], listay[62:], lw=5, label="Alta")
169 plt.axis([min(listax),max(listax),min(listay),max(listay)])
170 plt.grid()
171 print("Valor del area es >", listax[20],listay[20])
172 plt.legend(loc="lower right", title="Legend Title", frameon=False)
173 plt.axhline(resul[1], label='HORIZONTAL 2',color='black')
174 plt.axhline(resul[0], label='HORIZONTAL 1',color='blue')
175 plt.axvline(int(a), label='pyplot vertical line',color='red')
176
177 #SE GENERA EL AREA BAJO LA CURVA O INTERCECIONES
178 print("Valor de printe",listay[21:32])
179 listy = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]
180 print("VALOR ES >> ",a)
181

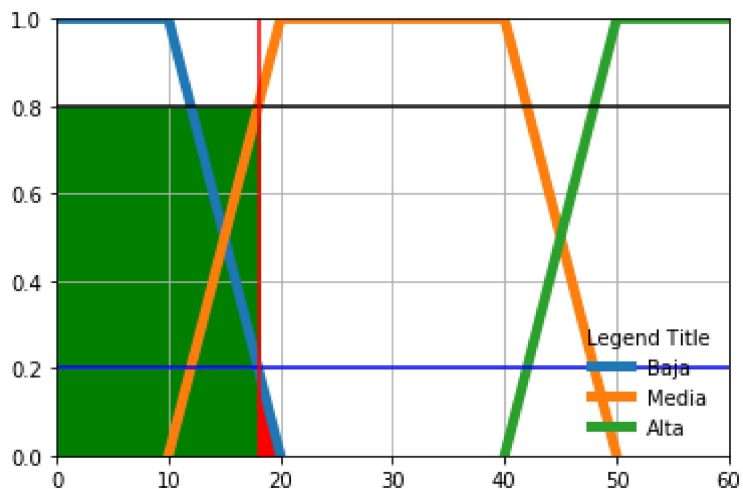
```

```

182 ly = []
183
184 for i in range(0,int(int(a)+1),1):
185     ly.append(resul[1])
186 print(ly)
187 print(listax[:int(int(a)+1)])
188 plt.fill_between(listax[:int(int(a)+1)],ly,color="green")
189 plt.fill_between(listax[int(a):21],listay[int(int(a)):21],color="red")
190 plt.show()
191 #SE CALCULA LAS VARIABLES CENTROIDES,AREAS PARCIALES,AREA TOTAL
192 #CALCULO DEL CENTROIDE
193 #CENTROIDE
194 c1 = int(a)/2
195 print("SE CALCULA LOS CENTROIDES")
196 print("VALOR DE C1",c1)
197 print(listax[20])
198 c2 = ((int(listax[20])-int(a))/3)+int(a)
199 print("VALOR DE C2 ",c2)
200 #AREAS PARCIALES
201 print("SE CALCULA LAS AREAS PARCIALES")
202 A1= a*float(resul[1])
203 print("Valor del area1 ",A1)
204 A2 = (a*float(resul[1]))/2
205 print("Valor de area2 ",A2)
206 #AREA TOTAL
207 at = float(A1+A2)
208 print("SE PRESENTA EL AREA TOTAL ",at)
209 #CALCULO DEL CENTROIDE
210 cc = (c1*A1+c2*A2)/at
211 print("VALOR DEL CALCULO DEL CENTROIDE ES >",cc)
212
213 boton_hum = Button(p1,text="HUMEDAD",command=grafico_hume)
214 boton_hum.pack()
215 root.mainloop()
216

```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]



In []: ▶

1

In [28]: ▶

1

In []: ▶

1

In []: ▶

1