

# Tarea 1.4 Ejercicio de Caracterización de un Universo Difuso

Abarca Romero José Ángel

Lógica Difusa

2TM9

Código de Python:

```
#Operaciones con conjuntos difusos
import numpy as np
import matplotlib.pyplot as plt

# Variables
x = np.arange(0, 1000, 0.1)                # Rango del universo
A = np.zeros(len(x))
B = np.zeros(len(x))
graficas = np.empty((3,len(x)))
funcionCampana = np.zeros(len(x))

#Funciones
def funcionTriangular(j):

    print("----- Función discreta triangular -----")
    a = int(input("Introduce el valor de la constante a: "))*10
    b = int(input("Introduce el valor de la constante b: "))*10
    c = int(input("Introduce el valor de la constante c: "))*10

    for i in range(len(x)):
        if i <= a:
            A[i] = 0
        elif a < i and i <= b:
            A[i] = (i - a)/(b - a)
        elif b < i and i <= c:
            A[i] = (c - i)/(c - b)
        elif c < i:
            A[i] = 0
```

```

graficas[j] = A

def funcionTrapezoidal(j):

    print("----- Función discreta trapezoidal -----")
    a = int(input("Introduce el valor de la constante a: "))*10
    b = int(input("Introduce el valor de la constante b: "))*10
    c = int(input("Introduce el valor de la constante c: "))*10
    d = int(input("Introduce el valor de la constante d: "))*10

    for i in range(len(x)):
        if i <= a:
            B[i] = 0
        elif a < i and i <= b:
            B[i] = (i - a)/(b - a)
        elif b < i and i <= c:
            B[i] = 1
        elif c < i and i <= d:
            B[i] = (d - i)/(d - c)
        elif d < i:
            B[i] = 0

    graficas[j] = B

def campana(j):
    print("----- Función continua de campana -----")
    a = float(input("Introduce el valor de la constante a: "))*10
    b = float(input("Introduce el valor de la constante b: "))*10
    c = int(input("Introduce el valor de la constante c: "))*10

    for i in range(len(x)):
        funcionCampana[i] = 1/(1 + abs((i-c)/a)**(2*b))

    if j > 2:
        plt.title("Función de campana")
        plt.plot(x,funcionCampana)
    else:
        graficas[j] = funcionCampana

def operaciones():
    for i in range(3):
        grafica = input(str(i) + " ¿Qué función desearía graficar (Triangular
[T] || Trapezoidal [P] || Campana [C])?: ")

```

```

if grafica == "T" or grafica == "t":
    funcionTriangular(i)
elif grafica == "P" or grafica == "p":
    funcionTrapezoidal(i)
elif grafica == "C" or grafica == "c":
    campana(i)

plt.figure(1)
plt.title("Niveles de estrés")
plt.ylabel("chi")
plt.xlabel("Unidades de Cambio Vital (UCV)")
plt.plot(x,graficas[0], x,graficas[1], x,graficas[2])

operaciones()
#campana(3)

#Graficación de funciones difusas discretas

```

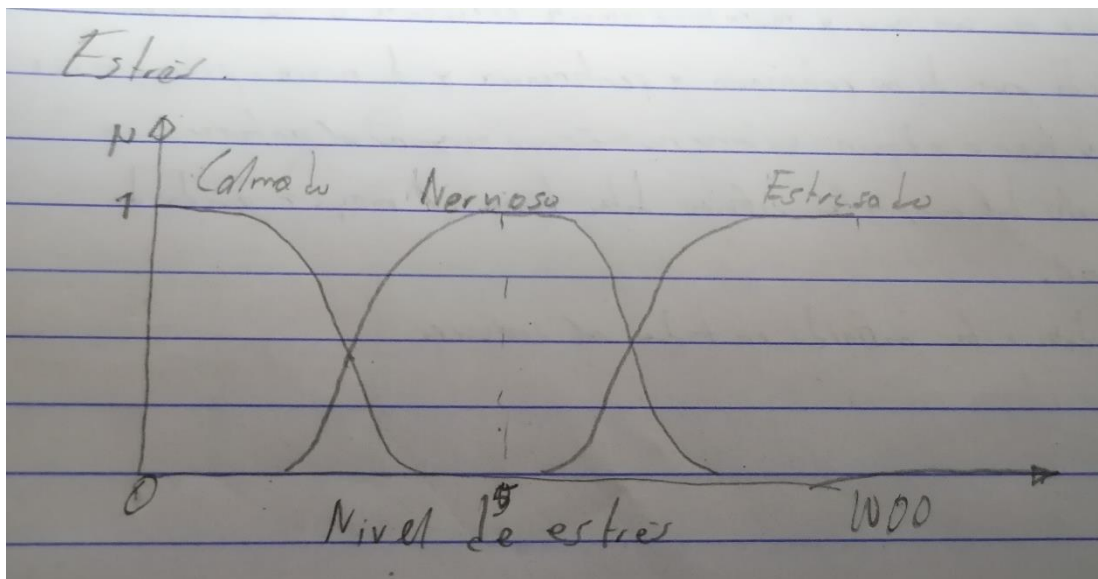
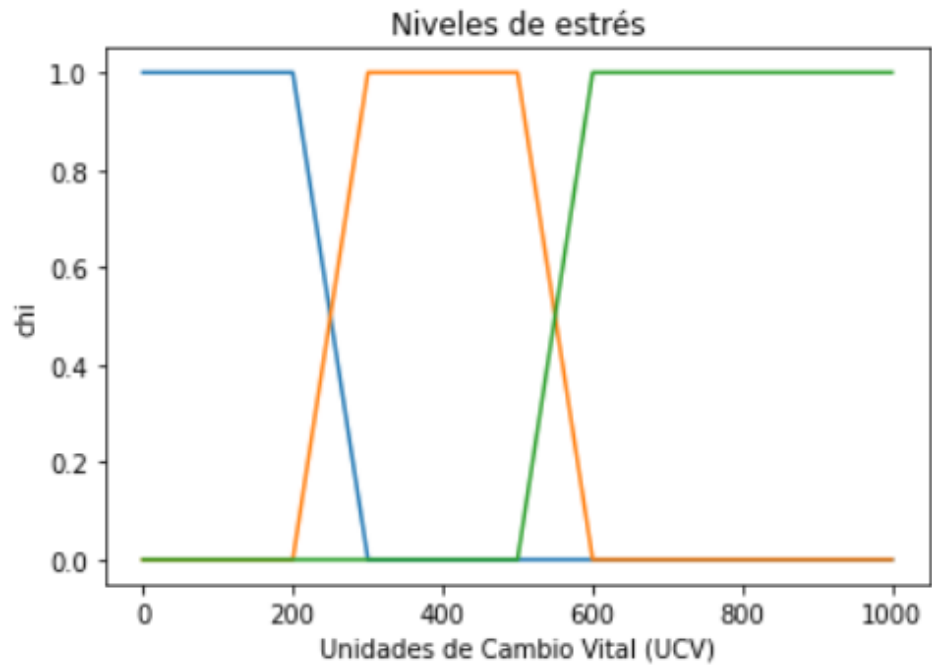
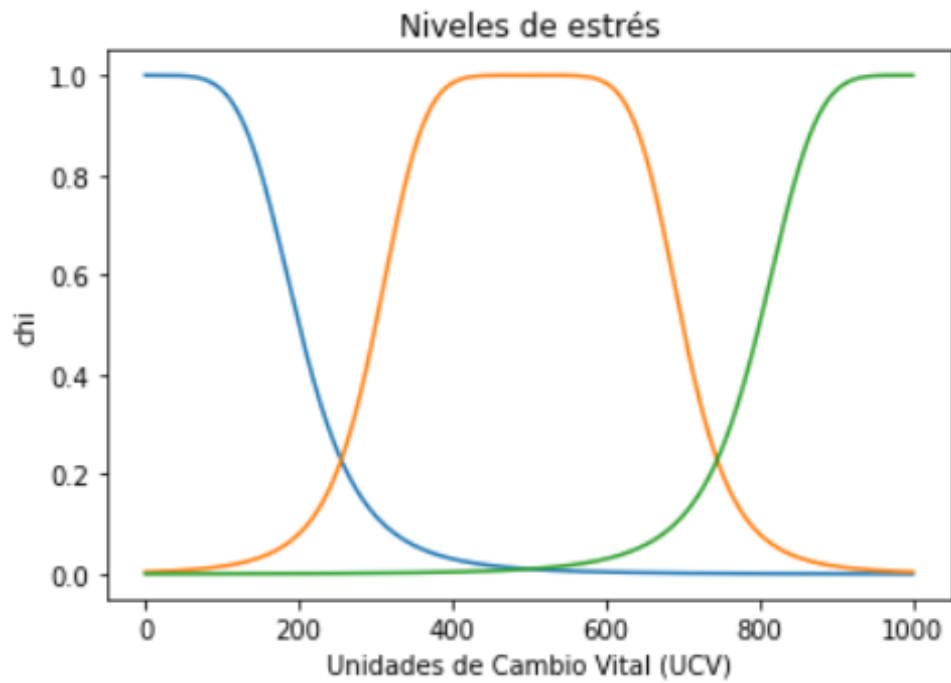


Ilustración 1 Bosquejo realizado en clase



*Ilustración 2 Caracterización de los niveles de estrés usando conjuntos discretos*



*Ilustración 3 Caracterización de los niveles de estrés usando conjuntos continuos*