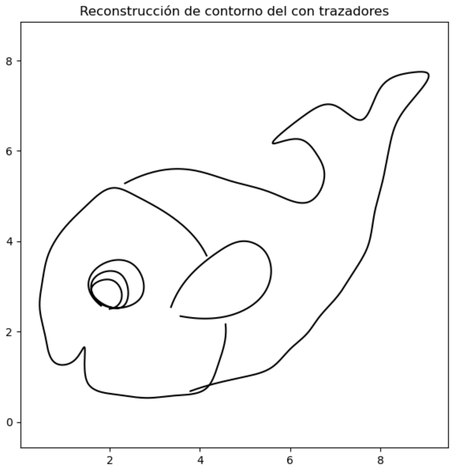
Hoja de Trabajo No.4

Métodos Numericos 1

José Rodrigo Oliveros González 22848

a. Use trazadores para graficar la figura dada.

In [9]:
```
import numpy as np
import scipy.interpolate as si
import matplotlib.pyplot as plt

xi1, yi1 = [4.15, 3.73, 3.23, 2.53, 2.09, 1.52, 0.98, 0.63, 0.51, 0.45, 0.58, 0.82, 1.23, 1.43, 1.51, 1.66, 2.11, 2.87, 3.42, 4.14, 4.4, 4.54, 4.57], [3.68, 4.25, 4.64, 5.03
xi2, yi2 = [3.36, 3.74, 4.4, 5.0, 5.4, 5.58, 5.42, 4.95, 4.25, 3.57], [2.54, 3.19, 3.76, 4.0, 3.82, 3.36, 2.84, 2.46, 2.29, 2.34]
xi3, yi3 = [2.34, 2.77, 3.46, 3.96, 4.69, 5.62, 6.44, 6.76, 6.58, 6.19, 5.63, 6.31, 7.0, 7.56, 8.0, 8.67, 9.03, 8.52, 8.23, 8.04, 7.87, 7.72, 7.39, 7.05, 6.63, 6.38, 6.0, 5.
xi4, yi4 = [1.81, 1.7, 2.5, 2.71, 1.71, 1.62, 2.23, 2.27, 1.6, 2.04, 2.27, 2.0], [2.57, 3.39, 3.5, 2.78, 2.69, 3.12, 3.28, 2.54, 2.96, 3.14, 2.84, 2.5]

# Lista de listas
segmentos = [(xi1, yi1),(xi2, yi2),(xi3, yi3),(xi4, yi4)]

def dibujar(xi, yi, npts=300):

    tck, u = si.splprep([xi, yi], s=0, k=3) # Parametrización de la curva
    x = np.linspace(0, 1, npts)
    out = si.splev(x, tck)
    return out

# Graficar
alto = 7
ancho = 7
plt.figure(figsize=(alto,ancho))

for (xi, yi) in segmentos:
    if len(xi) > 0 and len(yi) > 0: # Verificar que no estén vacíos
        spline_out = dibujar(xi, yi)
        if spline_out:
            plt.plot(spline_out[0], spline_out[1], "-", color="black")

plt.axis("equal")
plt.title("Reconstrucción de contorno del con trazadores")
plt.show()
```



b. Use el modelo generado para duplicar el tamaño de la gráfica. c. Use el modelo generado y su creatividad para girar la gráfica 45 grados.

In [11]:
```
import numpy as np
import scipy.interpolate as si
import matplotlib.pyplot as plt

xi1, yi1 = [4.15, 3.73, 3.23, 2.53, 2.09, 1.52, 0.98, 0.63, 0.51, 0.45, 0.58, 0.82, 1.23, 1.43, 1.51, 1.66, 2.11, 2.87, 3.42, 4.14, 4.4, 4.54, 4.57], [3.68, 4.25, 4.64, 5.03
xi2, yi2 = [3.36, 3.74, 4.4, 5.0, 5.4, 5.58, 5.42, 4.95, 4.25, 3.57], [2.54, 3.19, 3.76, 4.0, 3.82, 3.36, 2.84, 2.46, 2.29, 2.34]
xi3, yi3 = [2.34, 2.77, 3.46, 3.96, 4.69, 5.62, 6.44, 6.76, 6.58, 6.19, 5.63, 6.31, 7.0, 7.56, 8.0, 8.67, 9.03, 8.52, 8.23, 8.04, 7.87, 7.72, 7.39, 7.05, 6.63, 6.38, 6.0, 5.
xi4, yi4 = [1.81, 1.7, 2.5, 2.71, 1.71, 1.62, 2.23, 2.27, 1.6, 2.04, 2.27, 2.0], [2.57, 3.39, 3.5, 2.78, 2.69, 3.12, 3.28, 2.54, 2.96, 3.14, 2.84, 2.5]

# Lista de listas
segmentos = [(xi1, yi1),(xi2, yi2),(xi3, yi3),(xi4, yi4)]

def dibujar(xi, yi, npts=300):

    tck, u = si.splprep([xi, yi],s= 0,k=3) # Parametrización de la curva
    x= np.linspace(0, 1, npts)
    out = si.splev(x, tck)
    return out

# Matriz de rotación 45 grados
theta = np.deg2rad(45)
rot_matrix = np.array([[np.cos(theta), -np.sin(theta)],[np.sin(theta),  np.cos(theta)]])

# Graficar
alto = 7
ancho = 7
plt.figure(figsize=(2*alto,2*ancho)) # Duplicar tamaño

for (xi, yi) in segmentos:
    if len(xi) > 0 and len(yi) > 0: # Verificar que no estén vacíos
        spline_out = dibujar(xi, yi)
        if spline_out:
            coords = np.dot(rot_matrix, np.vstack((spline_out[0], spline_out[1]))) # Rotar
            x_rot, y_rot = coords[0], coords[1]
            plt.plot(x_rot, y_rot, "-", color="black")

plt.axis("equal")
plt.title("Reconstrucción de pez con trazadores, rotada 45° y tamaño duplicado")
plt.show()
```