

FiguraPizza

November 23, 2024

1 Código de Pizzache

1.0.1 Trabajo realizado por: Jessica Naomi Millan Sánchez

1.0.2 Graficación Computacional

1.0.3 Profesora: Hazem Álvarez Rodríguez

1.0.4 Clase del 16 de octubre de 2024

```
[7]: import matplotlib.pyplot as plt
import numpy as np
from matplotlib.patches import Arc, Rectangle, Circle, Polygon, Ellipse

# Crear una nueva figura
plt.figure(figsize=(10, 10))

# Dibujar un triángulo
triangle = plt.Polygon([[4, 4], [10, 16], [16, 4]], color="#FAB81A", alpha=.7,
    ↪label='Triángulo')
plt.gca().add_artist(triangle)

#Ovalo
oval = Ellipse((10, 4), width=12, height=2, color='#DA4C0B', label='Óvalo')
plt.gca().add_artist(oval)

#***** Lentes *****
# Dibujar un cuadrado con borde negro
Lente1 = plt.Rectangle((7, 10), 2, 2, color='black', edgecolor='black',
    ↪alpha=1, label='Cuadrado')
plt.gca().add_artist(Lente1)

# Dibujar un cuadrado
Lente2 = plt.Rectangle((11, 10), 2, 2, color='black', edgecolor='black',
    ↪alpha=1, label='Cuadrado')
plt.gca().add_artist(Lente2)
```

```

# Dibujar un rectángulo
rectangle = plt.Rectangle((9, 10.8), 2, 0.5, color='black', alpha=1,
    ↪label='Rectángulo')
plt.gca().add_artist(rectangle)

***** Peperonis *****
# --- Dibujar un Círculo ---
circle = plt.Circle((12.5, 8), 1, color='#D01D13', alpha=1, label='Círculo')
plt.gca().add_artist(circle)

circle = plt.Circle((7, 6), 1, color='#D01D13', alpha=1, label='Círculo')
plt.gca().add_artist(circle)

circle = plt.Circle((10.5, 13), 1, color='#D01D13', alpha=1, label='Círculo')
plt.gca().add_artist(circle)

***** Boca *****
center_x = 10.5 # Coordenada X del centro
center_y = 7.5  # Coordenada Y del centro
radius = 1      # Radio del medio círculo

# Crear un arreglo de puntos para el medio círculo
theta = np.linspace(np.pi, 0, 100) # Ángulo del medio círculo de 180 a 0 grados
x = center_x + radius * np.cos(theta)
y = center_y + radius * np.sin(theta)

# Convertir 195 grados a radianes
angle = 195 * np.pi / 180 # Ángulo en radianes

# Rotar los puntos
x_rotated = center_x + (x - center_x) * np.cos(angle) - (y - center_y) * np.
    ↪sin(angle)
y_rotated = center_y + (x - center_x) * np.sin(angle) + (y - center_y) * np.
    ↪cos(angle)

# Rellenar el medio círculo girado
plt.fill(x_rotated, y_rotated, color='black', alpha=1)

***** piernas *****
# Dibujar un rectángulo
rectangle = plt.Rectangle((7, .5), 1, 2.6, color='black', alpha=1,
    ↪label='Rectángulo')
plt.gca().add_artist(rectangle)

```

```

# Dibujar un rectángulo
rectangle = plt.Rectangle((12, .5), 1, 2.6, color='black', alpha=1,
    label='Rectángulo')
plt.gca().add_artist(rectangle)

0.8
# Configurar los límites del gráfico
plt.xlim(0, 20)
plt.ylim(0, 20)
plt.gca().set_aspect('equal', adjustable='box')

# Añadir leyenda
plt.legend()
plt.title('Figuras Geométricas en Python')
plt.grid(True)

# Mostrar la figura
plt.show()

```

<ipython-input-7-6d592cc554c1>:20: UserWarning: Setting the 'color' property will override the edgecolor or facecolor properties.

```
Lente1 = plt.Rectangle((7, 10), 2, 2, color='black', edgecolor='black',
alpha=1, label='Cuadrado')
```

<ipython-input-7-6d592cc554c1>:24: UserWarning: Setting the 'color' property will override the edgecolor or facecolor properties.

```
Lente2 = plt.Rectangle((11, 10), 2, 2, color='black', edgecolor='black',
alpha=1, label='Cuadrado')
```

