

MODULACION 8QAM

```

import numpy as np
import matplotlib.pyplot as plt

# Tabla de verdad 8-QAM basada en amplitud y fase
tabla_8qam = {
    '000': (0.765, -135),
    '001': (1.848, -135),
    '010': (0.765, -45),
    '011': (1.848, -45),
    '100': (0.765, 135),
    '101': (1.848, 135),
    '110': (0.765, 45),
    '111': (1.848, 45)
}

# Mostrar tabla
print("Tabla de verdad 8-QAM")
print(" QIC | Amplitud | Fase (°)")
print("-----")
for bits, (A, θ) in tabla_8qam.items():
    print(f" {bits} | {A:.3f} V | {θ:+}°")

# Parámetros
fs = 1000          # Frecuencia de muestreo
f_carrier = 10     # Frecuencia portadora (Hz)
T = 1 / f_carrier  # Período de símbolo
samples_per_symbol = fs
t_symbol = np.linspace(0, T, samples_per_symbol, endpoint=False)

# Secuencia de bits (todos los 8 símbolos)
secuencia = list(tabla_8qam.keys())

# Colores para distinguir cada símbolo
colores = plt.cm.tab10(np.linspace(0, 1, len(secuencia)))

# Generación de señal 8-QAM
señal_total = []
t_total = []
inicio_tiempo = 0

plt.figure(figsize=(16, 5))

for i, bits in enumerate(secuencia):
    A, fase = tabla_8qam[bits]
    θ_rad = np.radians(fase)
    portadora = A * np.cos(2 * np.pi * f_carrier * t_symbol + θ_rad)
    señal_total.extend(portadora)

    # Tiempo local del símbolo
    t_local = t_symbol + inicio_tiempo
    t_total.extend(t_local)
    inicio_tiempo += T

    # Dibujar cada símbolo con color diferente
    plt.plot(t_local, portadora, color=colores[i], label=f"{bits}")

plt.title("Señal modulada 8-QAM (colores por símbolo)")
plt.xlabel("Tiempo [s]")
plt.ylabel("Amplitud")
plt.grid(True)
plt.legend(title="Bits")
plt.tight_layout()
plt.show()

# Diagrama fasorial
plt.figure(figsize=(6, 6))
for bits, (A, fase) in tabla_8qam.items():
    θ_rad = np.radians(fase)
    I = A * np.cos(θ_rad)
    Q = A * np.sin(θ_rad)
    plt.arrow(0, 0, I, Q, head_width=0.1, length_includes_head=True)
    plt.text(I * 1.1, Q * 1.1, bits, fontsize=9, ha='center')

```

```
plt.axhline(0, color='gray', linewidth=0.5)
plt.axvline(0, color='gray', linewidth=0.5)
plt.title("Diagrama fasorial 8-QAM")
plt.xlabel("I (cos)")
plt.ylabel("Q (sen)")
plt.grid(True)
plt.axis('equal')
plt.tight_layout()
plt.show()
```

```
# Diagrama de constelación
plt.figure(figsize=(6, 6))
for bits, (A, fase) in tabla_8qam.items():
    theta_rad = np.radians(fase)
    I = A * np.cos(theta_rad)
    Q = A * np.sin(theta_rad)
    plt.plot(I, Q, 'ko')
    plt.text(I + 0.1, Q, bits, fontsize=9)
```

```
plt.axhline(0, color='gray', linewidth=0.5)
plt.axvline(0, color='gray', linewidth=0.5)
plt.title("Diagrama de constelación 8-QAM")
plt.xlabel("I (cos)")
plt.ylabel("Q (sen)")
plt.grid(True)
plt.axis('equal')
plt.tight_layout()
plt.show()
```



Tabla de verdad 8-QAM

QIC | Amplitud | Fase (°)

000	0.765 V	-135°
001	1.848 V	-135°
010	0.765 V	-45°
011	1.848 V	-45°
100	0.765 V	+135°
101	1.848 V	+135°
110	0.765 V	+45°
111	1.848 V	+45°

