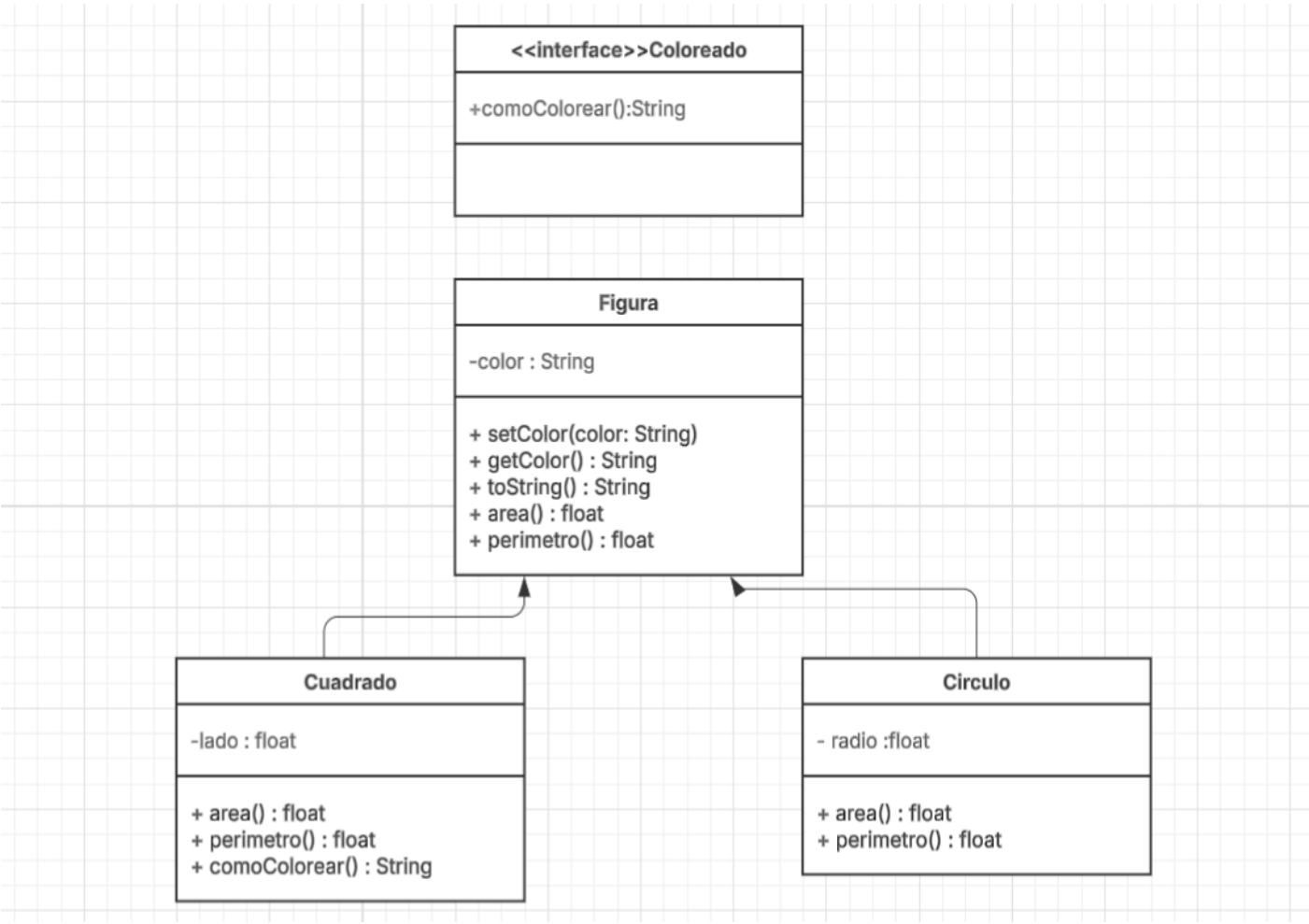
TEATRO MUNICIPAL

```
import tkinter as tk
from tkinter import ttk
from abc import ABC, abstractmethod


class Boleto(ABC):
    def __init__(self, numero):
        self.numero = numero

    @abstractmethod
    def get_precio(self):
        pass

    def __str__(self):
        return f"Número: {self.numero}, Precio: {self.get_precio():.2f}"


class Palco(Boleto):
    def __init__(self, numero):
        super().__init__(numero)

    def get_precio(self):
        return 100.00

class Platea(Boleto):
    def __init__(self, numero, dias_anticipacion):
        super().__init__(numero)
        self.dias_anticipacion = dias_anticipacion

    def get_precio(self):
        return 50.00 if self.dias_anticipacion >= 10 else 60.00

class Galeria(Boleto):
    def __init__(self, numero, dias_anticipacion):
        super().__init__(numero)
        self.dias_anticipacion = dias_anticipacion

    def get_precio(self):
        return 25.00 if self.dias_anticipacion >= 10 else 30.00


def vender_boleto():
    numero = int(entrada_numero.get())
    tipo = tipo_boleto.get()
    dias = entrada_dias.get()
    dias_anticipacion = int(dias) if dias else 0

    if tipo == "Palco":
        boleto = Palco(numero)
    elif tipo == "Platea":
        boleto = Platea(numero, dias_anticipacion)
    elif tipo == "Galería":
        boleto = Galeria(numero, dias_anticipacion)
    else:
        etiqueta_info.config(text="Por favor, selecciona un tipo válido.")
        return

    etiqueta_info.config(text=str(boleto))


def salir():
    ventana.destroy()
```

```
ventana = tk.Tk()
ventana.title("Teatro Municipal - Venta de Boletos")


tk.Label(ventana, text="Número de Boleto:").grid(row=0, column=0, padx=10, pady=5)
entrada_numero = tk.Entry(ventana)
entrada_numero.grid(row=0, column=1, padx=10, pady=5)

tk.Label(ventana, text="Cantidad de Días para el Evento:").grid(row=1, column=0, padx=10, pady=5)
entrada_dias = tk.Entry(ventana)
entrada_dias.grid(row=1, column=1, padx=10, pady=5)

tk.Label(ventana, text="Tipo de Boleto:").grid(row=2, column=0, padx=10, pady=5)
tipo_boleto = ttk.Combobox(ventana, values=["Palco", "Platea", "Galería"])
tipo_boleto.set("Selecciona un tipo")
tipo_boleto.grid(row=2, column=1, padx=10, pady=5)


boton_vender = tk.Button(ventana, text="Vender", command=vender_boleto)
boton_vender.grid(row=3, column=0, padx=10, pady=10)

boton_salir = tk.Button(ventana, text="Salir", command=salir)
boton_salir.grid(row=3, column=1, padx=10, pady=10)


etiqueta_info = tk.Label(ventana, text="Información del boleto aparecerá aquí.")
etiqueta_info.grid(row=4, column=0, columnspan=2, padx=10, pady=10)

ventana.mainloop()
```

COLOREADOS



```python
import tkinter as tk
from tkinter import ttk
from abc import ABC, abstractmethod
import random
import math


class Coloreado(ABC):
    @abstractmethod
    def comoColorear(self):
        pass


class Figura(ABC):
    def __init__(self, color):
        self.color = color

    def setColor(self, color):
        self.color = color

    def getColor(self):
        return self.color

    def __str__(self):
        return f"Color: {self.color}"

    @abstractmethod
    def area(self):
        pass

    @abstractmethod
    def perimetro(self):
        pass


class Cuadrado(Figura, Coloreado):
    def __init__(self, color, lado):
        super().__init__(color)
        self.lado = lado

    def area(self):
        return self.lado ** 2

    def perimetro(self):
        return 4 * self.lado

    def comoColorear(self):
        return "Colorear los cuatro lados"

# Clase Circulo
class Circulo(Figura):
    def __init__(self, color, radio):
        super().__init__(color)
        self.radio = radio

    def area(self):
        return math.pi * self.radio ** 2

    def perimetro(self):
        return 2 * math.pi * self.radio


def generar_figuras():
    figuras = []
    colores = ["Rojo", "Azul", "Verde", "Amarillo", "Morado"]
```

```
    for _ in range(5):
        tipo = random.randint(1, 2)
        color = random.choice(colores)

        if tipo == 1:
            lado = random.uniform(1.0, 10.0)
            figuras.append(Cuadrado(color, lado))
        elif tipo == 2:
            radio = random.uniform(1.0, 10.0)
            figuras.append(Circulo(color, radio))

    return figuras

def mostrar_figuras():
    figuras = generar_figuras()
    resultado.delete(1.0, tk.END)

    for figura in figuras:
        resultado.insert(tk.END, f"{figura}\n")
        resultado.insert(tk.END, f"Área: {figura.area():.2f}\n")
        resultado.insert(tk.END, f"Perímetro: {figura.perimetro():.2f}\n")
        if isinstance(figura, Coloreado):
            resultado.insert(tk.END, f"Cómo colorear: {figura.comoColorear()}\n"
        resultado.insert(tk.END, "\n")


ventana = tk.Tk()
ventana.title("Objetos Coloreados - Figuras")


etiqueta = tk.Label(ventana, text="Haz clic en el botón para generar 5 figuras a
etiqueta.pack(pady=10)


boton_generar = tk.Button(ventana, text="Generar Figuras", command=mostrar_figur
boton_generar.pack(pady=10)


resultado = tk.Text(ventana, height=15, width=50)
resultado.pack(pady=10)


boton_salir = tk.Button(ventana, text="Salir", command=ventana.destroy)
boton_salir.pack(pady=10)

ventana.mainloop()
```