



Alexandra Arisneidy Sanchez Santana

### Asignación: Ejercicios usando Herencia y Polimorfismo

1. Crea una clase base llamada **Animal** con un método `hablar()`. Luego crea clases hijas como **Perro** y **Gato** que sobrescriban el método.

```
class Animal:
    def hablar(self):
        print("El animal hace un sonido")

class Perro(Animal):
    def hablar(self):
        print("El perro dice: Guau")

class Gato(Animal):
    def hablar(self):
        print("El gato dice: Miau")

# Prueba
animales = [Perro(), Gato()]

for animal in animales:
    animal.hablar()
```

```
PS C:\Users\PC> & C:/Users/PC/
El perro dice: Guau
El gato dice: Miau
PS C:\Users\PC>
```

2. Crea una clase base **Empleado** con atributos nombre y salario. Crea clases hijas como Gerente y Técnico, cada una con un método `calcular_bono()` diferente.

```
class Empleado:
    def __init__(self, nombre, salario):
        self.nombre = nombre
        self.salario = salario

    def calcular_bono(self):
        return 0

class Gerente(Empleado):
    def calcular_bono(self):
        return self.salario * 0.20

class Tecnico(Empleado):
    def calcular_bono(self):
        return self.salario * 0.10

# Prueba
empleados = [
    Gerente("Alexandra", 50000),
    Tecnico("Juan", 30000)
]

for empleado in empleados:
    print(f"{empleado.nombre} - Bono: {empleado.calcular_bono()}")
El gato dice: Miau
PS C:\Users\PC> & C:/Users/PC/
Alexandra - Bono: 10000.0
Juan - Bono: 3000.0
PS C:\Users\PC>
```

3. Crea una clase base **Figura** con un método `area()`. Implementa clases hijas como Círculo y Cuadrado que calculen el área según corresponda.

```
import math

class Figura:
```

```

    def area(self):
        return 0

class Circulo(Figura):
    def __init__(self, radio):
        self.radio = radio

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

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

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

# Prueba
figuras = [Circulo(5), Cuadrado(4)]

for figura in figuras:
    print(f"Área: {figura.area()}")
PS C:\Users\PC> & C:/Users/f
Área: 78.53981633974483
Área: 16
PS C:\Users\PC>

```

4. Crea una clase **Vehiculo** con un método `mover()`. Crea clases hijas como **Carro** y **Bicicleta** que implementen su propia versión del método.

```

class Vehiculo:
    def mover(self):
        print("El vehículo se mueve")

class Carro(Vehiculo):
    def mover(self):
        print("El carro se mueve con motor")

class Bicicleta(Vehiculo):
    def mover(self):
        print("La bicicleta se mueve pedaleando")

# Prueba
vehiculos = [Carro(), Bicicleta()]

for v in vehiculos:

```

```
v.mover()  
PS C:\Users\PC> & C:/Users/PC/AppData  
El carro se mueve con motor  
La bicicleta se mueve pedaleando  
PS C:\Users\PC>
```

5. Crea una clase **Dispositivo** con un método `encender()`. Crea clases hijas como Laptop y Teléfono que sobrescriban el comportamiento del método.

```
class Dispositivo:  
    def encender(self):  
        print("El dispositivo se está encendiendo")  
  
class Laptop(Dispositivo):  
    def encender(self):  
        print("La laptop está encendida")  
  
class Telefono(Dispositivo):  
    def encender(self):  
        print("El teléfono está encendido")  
  
# Prueba  
dispositivos = [Laptop(), Telefono()]  
  
for d in dispositivos:  
    d.encender()  
PS C:\Users\PC> & C:/Users/PC/AppData  
La laptop está encendida  
El teléfono está encendido  
PS C:\Users\PC>
```