

Universidad Nacional de Colombia

Sede Medellín

Facultad de Ciencias
Departamento de Estadística



Curso: Modelos No Lineales

Informe de Lectura
Modelos Lineales Generalizados

Profesor:
Victor Ignacio Lopez Rios

Estudiante:
David Esteban Cartagena Mejia

Fecha de presentación: 3 de mayo de 2025

Año Académico 2025-I

Tabla de contenidos

Ejercicio 2.1	3
ejercicio 2.2	5

Listado de Figuras

1 Diagrama UML de la clase Persona	3
2 Planeta Tierra	5

Listado de Tablas

Ejercicio 2.1

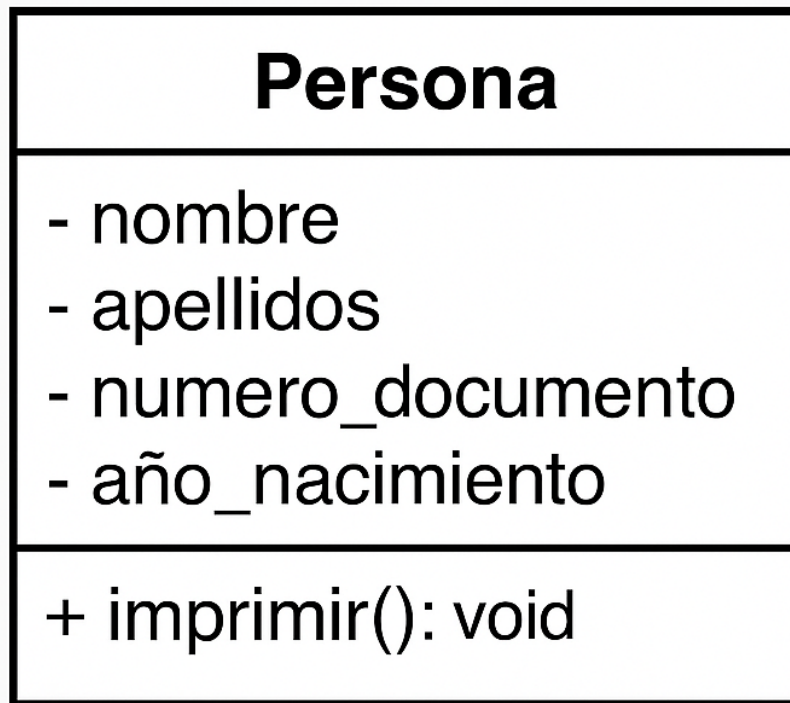


Figura 1: Diagrama UML de la clase Persona

```
class Persona:
    def __init__(self, nombre,
                 apellidos, numero_documento, ano_nacimiento):
        self.nombre = nombre
```

```

        self.apellidos = apellidos
        self.numero_documento = numero_documento
        self.ano_nacimiento = ano_nacimiento

    def imprimir(self):
        print(f"Nombre = {self.nombre}")
        print(f"Apellidos = {self.apellidos}")
        print(f"Número de documento de identidad = {self.numero_documento}")
        print(f"Año de nacimiento = {self.ano_nacimiento}")
        print()

# Método main simulado
if __name__ == "__main__":
    p1 = Persona("Pedro", "Pérez", "1053121010", 1998)
    p2 = Persona("Luis", "León", "1053223344", 2001)

    p1.imprimir()
    p2.imprimir()

```

Nombre = Pedro
 Apellidos = Pérez
 Número de documento de identidad = 1053121010
 Año de nacimiento = 1998

Nombre = Luis
 Apellidos = León
 Número de documento de identidad = 1053223344
 Año de nacimiento = 2001

ejercicio 2.2

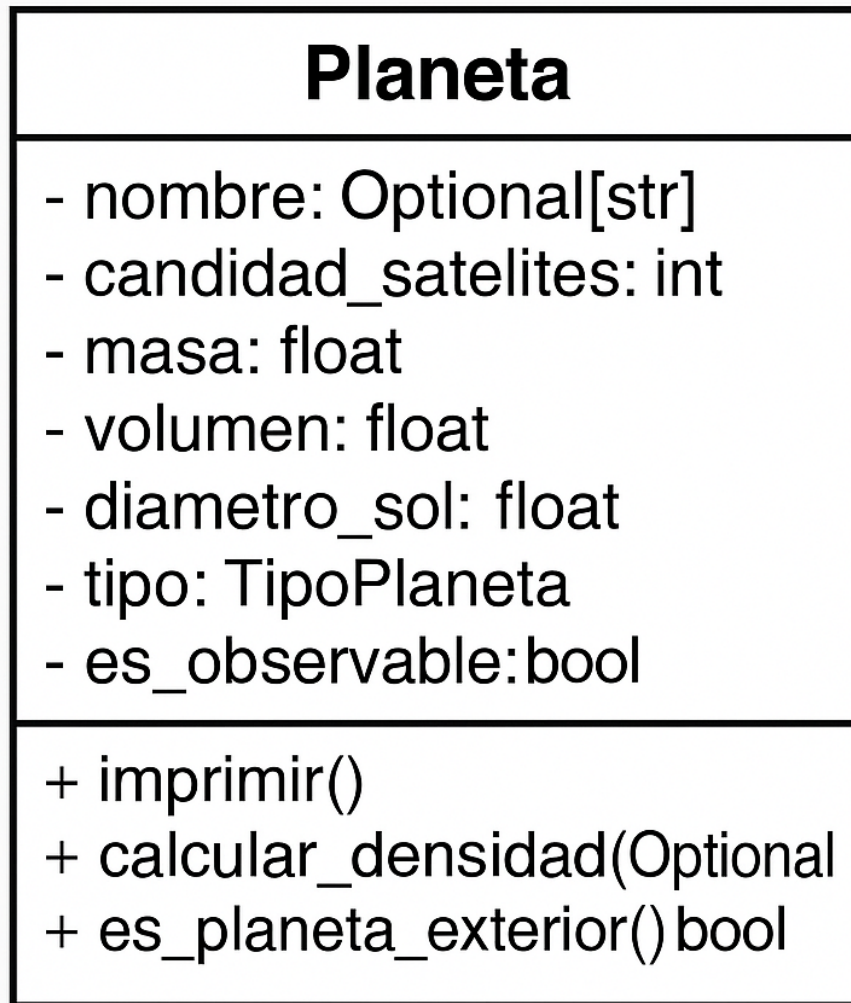


Figura 2: Planeta Tierra

```
from enum import Enum

class TipoPlaneta(Enum):
    GASEOSO = "GASEOSO"
```

```

TERRESTRE = "TERRESTRE"
ENANO = "ENANO"

class Planeta:
    def __init__(self, nombre=None, cantidad_satelites=0, masa=0, volumen=0,
d diametro=0, distancia_sol=0, tipo=TipoPlaneta.TERRESTRE, es_observable=False):
        self.nombre = nombre
        self.cantidad_satelites = cantidad_satelites
        self.masa = masa
        self.volumen = volumen
        self.diametro = diametro
        self.distancia_sol = distancia_sol
        self.tipo = tipo
        self.es_observable = es_observable

    def imprimir(self):
        print(f"Nombre del planeta = {self.nombre}")
        print(f"Cantidad de satélites = {self.cantidad_satelites}")
        print(f"Masa del planeta = {self.masa}")
        print(f"Volumen del planeta = {self.volumen}")
        print(f"Diámetro del planeta = {self.diametro}")
        print(f"Distancia al Sol = {self.distancia_sol}")
        print(f"Tipo de planeta = {self.tipo.value}")
        print(f"Es observable = {self.es_observable}")

    def calcular_densidad(self):
        if self.volumen == 0:
            return None
        return self.masa / self.volumen

    def es_planeta_exterior(self):
        UA = 149_597_870 # km
        limite_exterior = 3.4 * UA
        return self.distancia_sol > limite_exterior

if __name__ == "__main__":
    tierra = Planeta("Tierra", 1, 5.9736e24, 1.08321e12, 12742, 150_000_000, TipoPlaneta.T
    for planeta in [tierra]:
        planeta.imprimir()
        densidad = planeta.calcular_densidad()
        print(f"Densidad del planeta = {densidad:.2f} kg/km³" if densidad else "Densidad n
        print(f"Es planeta exterior = {planeta.es_planeta_exterior()}")

```

```
print()
```

```
Nombre del planeta = Tierra  
Cantidad de satélites = 1  
Masa del planeta = 5.9736e+24  
Volumen del planeta = 1083210000000.0  
Diámetro del planeta = 12742  
Distancia al Sol = 150000000  
Tipo de planeta = TERRESTRE  
Es observable = True  
Densidad del planeta = 5514720137369.48 kg/km3  
Es planeta exterior = False
```