

Trabajo de POO

Richard Beycker Herkt Pilataxi

Pontificia Universidad Católica del Ecuador

Sede Ambato

Ing. Edison Fernando Meneses Torres

Programación II

Tercer Semestre

15 de abril de 2025

Ambato

Av. Manuelita Saenz

0986551827

rbherkt@pucesa.edu.ec

1. Ejercicio 1: Crear una clase Libro Crea una clase que tenga los atributos: *título, autor, anio*. Agrega un método que imprima un mensaje con los datos del libro.

```
class Libro:
    def __init__(self,titulo,autor,anio):
        self.titulo=titulo
        self.autor=autor
        self.anio=anio

    def descripcion(self):
        return f"El nombre es {self.titulo}, el tema es {self.autor} y el año es {self.anio}"

libro1=Libro("Pepito y sus amigos ", "Richard Herkt", 2020)
print(libro1.descripcion())
```

```
El nombre es Pepito y sus amigos , el tema es Richard Herkt y el año es 2020
PS C:\3TH SEMESTER\RDA1\PROGRAMACION 2>
```

- **Descripción:**

Este ejercicio define una clase llamada 'Libro' con los atributos: título, autor y anio de publicación.

Creamos el método descripción, que imprime un mensaje con los datos del libro. Esto nos permite interactuar con los atributos de manera sencilla y amigable.

2. Ejercicio 2: Clase Estudiante:

Crea una clase con atributos nombre, carrera, y nota_final. Agrega un método que diga si aprobó ($\text{nota} \geq 7$).

```
class estudiante:
    def __init__(self,nombre,carrera,nota_final):
        self.nombre=nombre
        self.carrera=carrera
        self.nota_final=nota_final

    def presen(self):
        return f"Hola! mi nombre es {self.nombre}, sigo la carrera de {self.carrera} y mi nota final es {self.nota_final}"

    def veri(self):
        if self.nota_final>=7:
            return "Estoy Aprobado "
        else:
            return "Reprobado "

personal=estudiante(input("Ingrese su nombre.."),input("Ingrese su carrera.."),float(input('Ingrese su nota final..')))

print(personal.presen())
print(personal.veri())
```

```
Ingrese su nombre..Richard
Ingrese su carrera..Ingenieria Civil
Ingrese su nota final..7
Hola! mi nombre es Richard, sigo la carrera de Ingenieria Civil y mi nota final es 7.0
Estoy Aprobado
PS C:\3TH SEMESTER\RDA1\PROGRAMACION 2> |
```

- **Descripción:**
- Este ejercicio define una clase *Estudiante* que almacena nombre, carrera y nota final. Tiene métodos para mostrar la información del estudiante y determinar si aprobó o no.
- Se utilizaron métodos para presentar los datos del estudiante de manera clara y verificar si su nota final es suficiente para aprobar (igual o superior a 7).

3. Ejercicio 3: Herencia Crea una clase base Vehiculo con el método moverse. Crea una clase Auto que herede de Vehiculo y sobrescriba el método:

```
class Vehiculo:
    def moverse(self):
        print("El vehiculo se mueve")

class Auto(Vehiculo):
    def moverse(self):
        print("El auto se mueve ")

vehiculo=Vehiculo()

vehiculo.moverse()
auto=Auto()
auto.moverse()
```

```
El vehiculo se mueve
El auto se mueve
PS C:\3TH SEMESTER\RDA1\PROGRAMACION 2>
```

- **Descripción:**

Se define la clase Vehiculo con un método moverse que representa una acción genérica de movimiento.

La clase Auto hereda de la clase base Vehiculo, lo cual significa que Auto puede usar o modificar los métodos de Vehiculo.

4. Ejercicio 4: Polimorfismo Crea dos clases: Pajaro y Gato, cada una con el método sonido(). Luego, usa una función que reciba un objeto y llame al método sonido():

```
class Pajaro:
    def sonido(self):
        print("El pajaro hace un sonido")

class Gato:
    def sonido(self):
        print("El gato hace un sonido")

def hacer_sonido(animal):
    animal.sonido()

gato=Gato()
pajaro=Pajaro()
hacer_sonido(pajaro)

hacer_sonido(gato)
```

```
El pajarero hace un sonido  
El gato hace un sonido  
PS C:\3TH SEMESTER\RDA1\PROGRAMACION 2>
```

- **Descripción:**

Este ejercicio muestra polimorfismo en Python. Se crean dos clases (Pajaro y Gato), cada una con un método sonido(). Luego, la función hacer_sonido(animal) recibe un objeto y llama su método sonido(), sin importar su tipo. Esto permite reutilizar código sin modificar la función.