

Untitled

Práctica 2. Clases y Objetos, métodos y atributos

```
class Persona: def init(self, nombre, apellido, edad): # Creación de la clase self.nombre = nombre self.apellido = apellido self.edad = edad self.__cuenta = None # Atributo privado
```

```
def asignar_cuenta(self, cuenta):
    self.__cuenta = cuenta
    print(f"{self.nombre} ahora tiene una cuenta bancaria")

def consultar_saldo(self):
    if self.__cuenta:
        print(f"El saldo de {self.nombre} es:
${self.__cuenta.mostrar_saldo()}")
    else:
        print(f"{self.nombre} no tiene una cuenta bancaria asignada.")

# Nuevos métodos para operaciones bancarias
def depositar_dinero(self, cantidad):
    if self.__cuenta:
        self.__cuenta.depositar(cantidad)
    else:
        print(f"{self.nombre} no tiene una cuenta bancaria asignada.")

def retirar_dinero(self, cantidad):
    if self.__cuenta:
        self.__cuenta.retirar(cantidad)
    else:
        print(f"{self.nombre} no tiene una cuenta bancaria asignada.")

def presentarse(self):
    print(f"Hola, mi nombre es {self.nombre} {self.apellido} y tengo
{self.edad} años.")

def cumplir_anios(self):
    self.edad += 1
    print(f"¡Feliz cumpleaños! {self.nombre} ahora tiene {self.edad}
años.")
```

```
class CuentaBancaria: def init(self, numero_cuenta, saldo): self.numero_cuenta = numero_cuenta self.__saldo = saldo # Atributo privado
```

```
def mostrar_saldo(self):
    return self.__saldo

def depositar(self, cantidad):
```

```

        if cantidad > 0:
            self.__saldo += cantidad
            print(f"Se han depositado ${cantidad} a la cuenta. Nuevo saldo:
${self.__saldo}")
        else:
            print("La cantidad a depositar debe ser positiva.")

def retirar(self, cantidad):
    if 0 < cantidad <= self.__saldo:
        self.__saldo -= cantidad
        print(f"Se han retirado ${cantidad} de la cuenta. Nuevo saldo:
${self.__saldo}")
    else:
        print("Fondos insuficientes o cantidad inválida.")

```

Creación de objetos

```

estudiante1 = Persona("Juan", "Perez", 20) estudiante2 = Persona("Maria", "Gomez", 22)
cuenta1 = CuentaBancaria("0001", 1000)

```

Operaciones corregidas

```

print("=== OPERACIONES BANCARIAS ===") estudiante1.presentarse()
estudiante1.asignar_cuenta(cuenta1) estudiante1.consultar_saldo()
estudiante1.depositar_dinero(500) # Ahora funciona correctamente
estudiante1.retirar_dinero(200) # Ahora funciona correctamente
estudiante1.consultar_saldo()

print("") estudiante2.presentarse() estudiante2.consultar_saldo() # Mostrará que no tiene
cuenta

print("") estudiante1.cumplir_anios() estudiante1.presentarse()

```

EJERCICIO 1. Clase Coche corregida

```

print("=== OPERACIONES CON COCHES ===")

class Coche:
    def __init__(self, marca, modelo, anio):
        self.marca = marca
        self.modelo = modelo
        self.anio = anio
        self.velocidad = 0 # Inicializar velocidad aquí

    def datos(self):
        print(f"El coche es {self.marca} {self.modelo} del año {self.anio}.")

    def frenar(self, decremento):
        if decremento > self.velocidad:
            self.velocidad = 0

```

```

        else:
            self.velocidad -= decremento
            print(f"El coche {self.marca} {self.modelo} ha frenado a {self.velocidad} km/h.")

def acelerar(self, incremento):
    if incremento > 0:
        self.velocidad += incremento
        print(f"El coche {self.marca} {self.modelo} ha acelerado a {self.velocidad} km/h.")
    else:
        print("El incremento debe ser positivo.")

def estado_velocidad(self):
    print(f"Velocidad actual: {self.velocidad} km/h")

```

Crear objetos Coche

```

automovil1 = Coche("Toyota", "Corolla", 2020) automovil2 = Coche("Honda", "Civic", 2019)

automovil1.datos() automovil2.datos()

print("— Pruebas de velocidad —") automovil1.acelerar(120) automovil1.frenar(70)
automovil1.estado_velocidad()

print("") automovil2.acelerar(150) automovil2.frenar(90) automovil2.estado_velocidad()

```