

Practica 2

Practica 2. Clases, Objetos, Metodos y Atributos

```
class Persona: def __init__(self, nombre, apellido, edad): # Creacion de atributos 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 creada.")

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

def cumplir_anios(self):
    self.edad += 1
    print(f"Esta persona cumplio: {self.edad} años")

class cuenta_bancaria: def __init__(self, numero_cuenta, saldo): self.numero_cuenta = numero_cuenta self.saldo = saldo #Dato / Atributo Privado def mostrar_saldo(self): return self.saldo def depositar(self, cantidad): if cantidad > 0: self.__saldo += cantidad print(f"Se deposito la cantidad de ${self.cantidad} a la cuenta") else: print("Ingresa la cantidad valida")
def retirar(self, cantidad): if cantidad > 0 and cantidad <= self.saldo: self.saldo -= cantidad print(f"Se retiro la cantidad de ${self.cantidad} de la cuenta") else: print("Cantidad invalida o saldo insuficiente")
```

Creacion del objeto o instancia de una clase

```
estudiante1 = Persona("Alondra", "Gonzalez", 18) estudiante2 = Persona("Emi", "Solis", 19)
```

```
estudiante1.presentarse() #Mandar llamar los metodos(acciones) estudiante2.presentarse() estudiante1.cumplir_anios()
```

```
cuenta1 = cuenta_bancaria("001", 500) estudiante1.asignar_cuenta(cuenta1) # Se esta haciendo la relacion estudiante1.consultar_saldo()
```