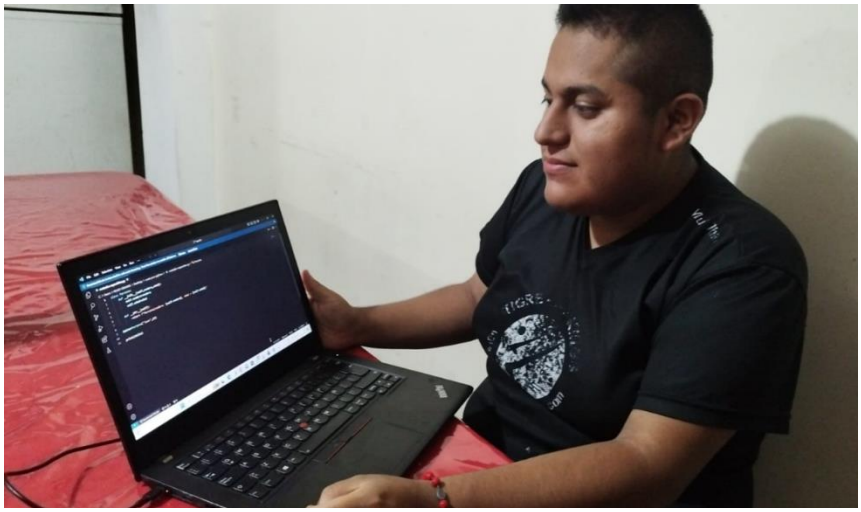
C: > Users > HUGO GRANDE > Desktop > curso poo python > ejercicio 3.py > Personaje

```python
class Personaje:
    def __init__(self,nombre,fuerza,velocidad):
        self.nombre=nombre
        self.fuerza=fuerza
        self.velocidad=velocidad
    def __repr__(self):
        return f"{self.nombre} (Fuerza : {self.fuerza}, velocidad: {self.velocidad})"

    def __add__(self, otro_pj):
        nuevo_nombre= self.nombre + "-" + otro_pj.nombre
        nuevo_fuerza= ((self.fuerza + otro_pj.fuerza)/2)**2
        nuevo_velocidad=((self.velocidad + otro_pj.velocidad)/2)**2
        return Personaje(nuevo_nombre,nuevo_fuerza,nuevo_velocidad)

goku=Personaje("Goku",12222,331313)
vegeta=Personaje("Vegeta",11000,20010)

gogeta = goku + vegeta
print(gogeta)
```

C: > Users > HUGO GRANDE > Desktop > curso poo python > metodos especiales.py

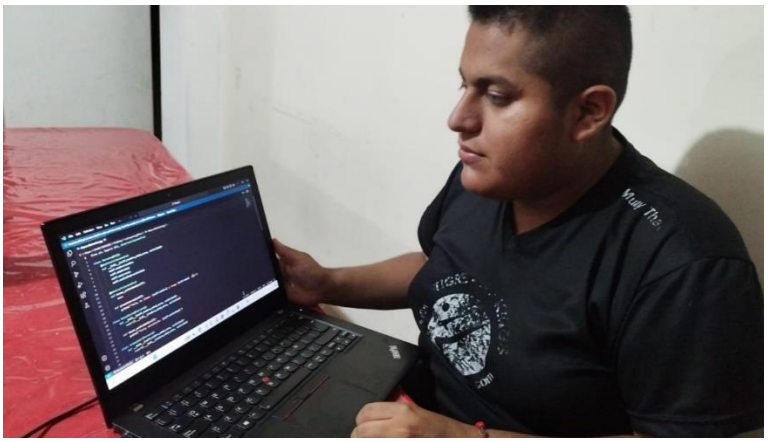
```python
class Persona:
    def __init__(self,nombre,edad):
        self.nombre=nombre
        self.edad=edad

    def __str__(self):
        return f"Persona(nombre: {self.nombre}, edad = {self.edad})"

dalto=Persona("Juan",23)

print(dalto)
```



C: > Users > HUGO GRANDE > Desktop > curso poo python > clases abstractas.py > ...

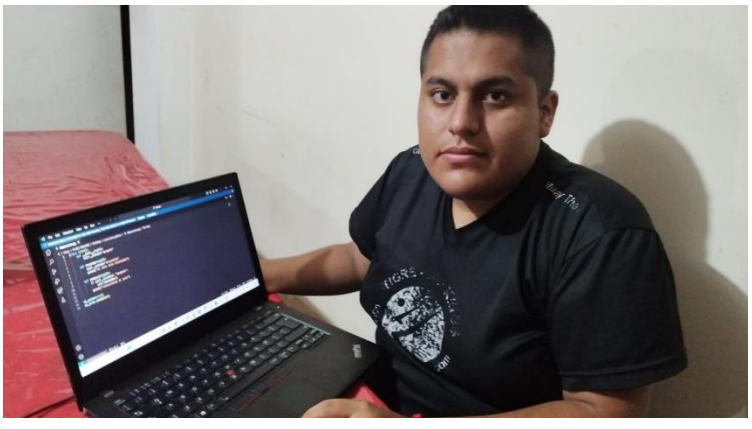
```python
from abc import ABC, abstractclassmethod

class Persona(ABC):
    @abstractclassmethod
    def __init__(self,nombre,edad,sexo, actividad):
        self.nombre=nombre
        self.edad=edad
        self.sexo=sexo
        self.actividad=actividad

    @abstractclassmethod
    def hacer_actividad(self):
        pass

    def presentarse(self):
        print(f"Hola, me llamo: {self.nombre} y tengo {self.edad}  años")

class Estudiante(Persona):
    def __init__(self,nombre,edad,sexo, actividad):
        super().__init__(nombre,edad,sexo, actividad)

    def hacer_actividad(self):
        print(f"Estoy estudiando: {self.actividad}")

class trabajador(Persona):
    def __init__(self,nombre,edad,sexo, actividad):
        super().__init__(nombre,edad,sexo, actividad)

```

Restricted Mode ⊗ 0 ⚠ 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Pyth



C: > Users > HUGO GRANDE > Desktop > curso poo python > Abstraccion.py

```python
class Auto():
    def __init__(self):
        self._estado="apagado"

    def encender(self):
        self._estado="encendido"
        print("el auto esta encendido")

    def conducir(self):
        if self._estado== "apagado":
            self.encender()
        print("conduciendo el auto")

mi_auto=Auto()
mi_auto.conducir()
```

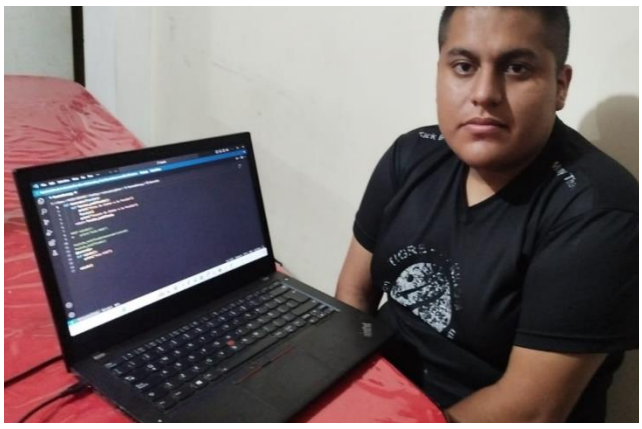


C: > Users > HUGO GRANDE > Desktop > curso poo python > Propiedades.py

```python
class Persona:
    def __init__(self,nombre,edad):
        self.__nombre=nombre
        self.__edad=edad

    @property
    def nombre(seft):
        return seft.__nombre



dalton=Persona("Lucas",21)

nombre= dalton.nombre
print(nombre)

```

C: > Users > HUGO GRANDE > Desktop > curso poo python > Decoradores.py

```python
def decorador(funcion):
    def funcion_modificada():
        print("Antes de llamar a la funcion")
        funcion()
        print("Despues de llamar a la funcion")
    return funcion_modificada

#def saludo():
#    print("Hola MADI")

#saludo_modificado=decorador(saludo)
#saludo_modificado()
@decorador
def saludo():
    print("Hola MADI")

saludo()
```

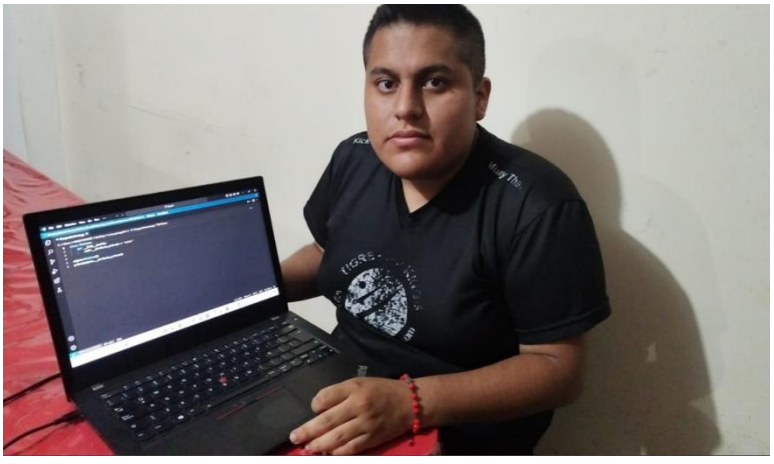


C: > Users > HUGO GRANDE > Desktop > curso poo python > Getters y Setters.py

```python
class Persona:
    def __init__(self,nombre,edad):
        self.__nombre=nombre
        self.__edad=edad

    def get_nombre(seft):
        return seft.__nombre

    def set_nombre(seft,new_nombre):
        seft.__nombre=new_nombre


dalton=Persona("Lucas",21)

nombre= dalton.get_nombre()
print(nombre)

dalton.set_nombre("Madisson")

nombre= dalton.get_nombre()
print(nombre)

```

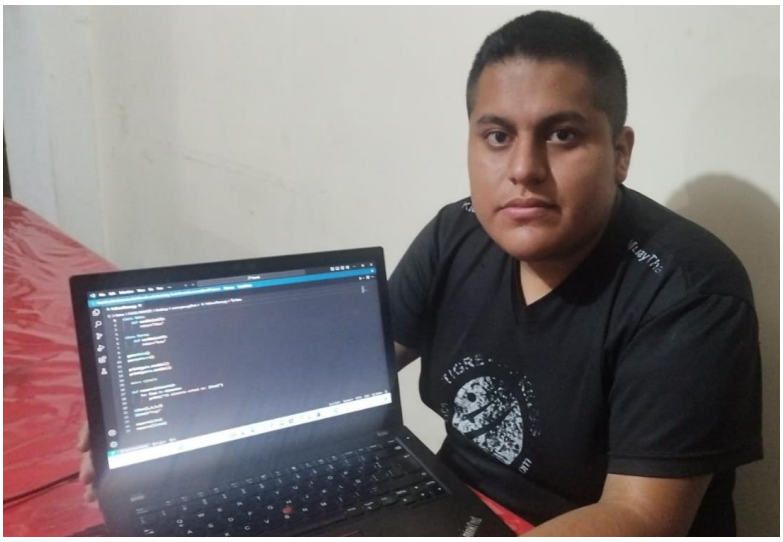
C: > Users > HUGO GRANDE > Desktop > curso poo python > Encapsulamiento.py

```python
class Miclase:
    def __init__(self):
        self.__atributo_privado = "Valor"

objeto=Miclase()
print(objeto.__atributo_privado)
```



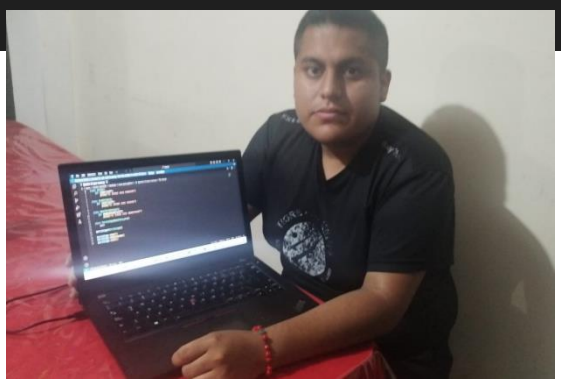
C: > Users > HUGO GRANDE > Desktop > curso poo python > Polimorfismo.py

```python
class Gato:
    def sonido(self):
        return"Miau"

class Perro:
    def sonido(self):
        return"Guau"

gato=Gato()
perro=Perro()

print(gato.sonido())
print(perro.sonido())

#otro ejemplo

def recorre(elemento):
    for item in elemento:
        print(f"El elemento actual es: {item}")

Lita={1,2,3,4}
lista2="Hugo"

recorre(Lita)
recorre(lista2)
```



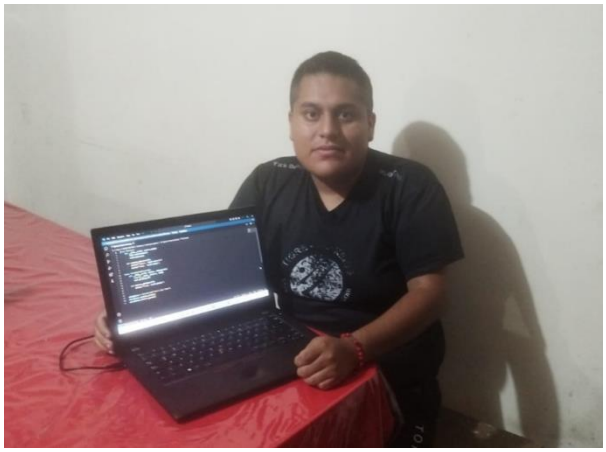
> Users > HUGO GRANDE > Desktop > curso poo python > Ejecicio 2.2 poo curso.py

```python
class Animal:
    def comer(self):
        print("El animal esta comiendo")

class Ave(Animal):
    def volar(self):
        print("El animal esta volando")

class Mamifero(Animal):
    def amamantando(self):
        print("El animal esta amamantando")

class Murcielago(Mamifero,Ave):
    pass

murcielago=Murcielago()

murcielago.comer()
murcielago.amamantando()
murcielago.volar()
```



C: > Users > HUGO GRANDE > Desktop > curso poo python > Ejecicio 2 poo curso.py > Persona

```python
class Persona:
    def __init__(self, nombre,edad):
        self.nombre=nombre
        self.edad=edad

    def mostrar_datos(self):
        print(f"Nombre: {self.nombre}")
        print(f"Edad:  {self.edad}")

class Estudiante(Persona):
    def __init__(self, nombre, edad,grado):
        super().__init__(nombre, edad)
        self.grado=grado

    def mostrar_grado(self):
        print(f"Grado: {self.grado}")


estudiante = Estudiante("Pedro",13,"10mo")
estudiante.mostrar_datos()
estudiante.mostrar_grado()
```



C: > Users > HUGO GRANDE > Desktop > curso poo python > mro.py > A

```python
class A:
    def hablar(self):
        print("Hola desde A")

class B(A):
    def hablar(self):
        print("Hola desde B")

class C(A):
    def hablar(self):
        print("Hola desde C")

class D(B,C):
    def hablar(self):
        print("Hola desde D")

d=D()
d.hablar()

B.hablar(d)
C.hablar(d)
A.hablar(d)

print(D.mro())
```

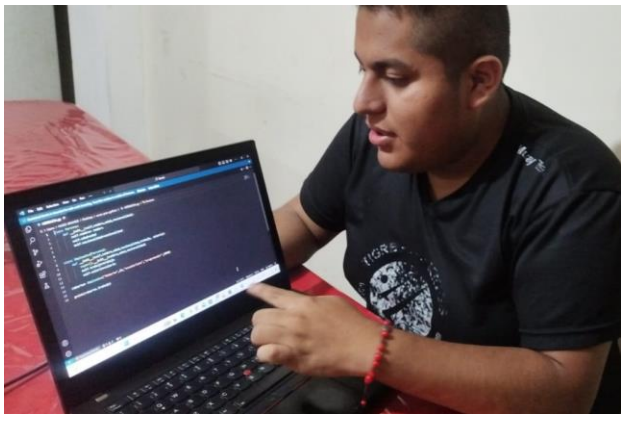
> Users > HUGO GRANDE > Desktop > curso poo python > Herencia multiple.py > Persona

```python
class Persona:
    def __init__(self,nombre,edad,nacionalidad):
        self.nombre= nombre
        self.edad=edad
        self.nacionalidad=nacionalidad
    def hablar(self):
        print("Hola,estoy heblando mucho")

class Artista:
    def __init__(self,habilidad):
        self.habilidad=habilidad

    def mostrar_habilidad(self):
        return f"Mi habilidad es: {self.habilidad}"

class EmpleadoArtista(Persona,Artista):
    def __init__(self, nombre, edad, nacionalidad,habilidad,salario,empresa):
        Persona.__init__(self,nombre, edad, nacionalidad)
        Artista.__init__(self,habilidad)
        self.salario=salario
        self.empresa=empresa


    def presentarse(self):
        print(f"Hola, soy: {self.nombre}, {self.mostrar_habilidad()} y trabajo en {self.empresa}")

roberto= EmpleadoArtista("Roberto",23,"ecuatoriano","cantar",1000,"pizza hat")

```

stricted Mode ⊗ 0 ⚠ 0 0 Ln 1

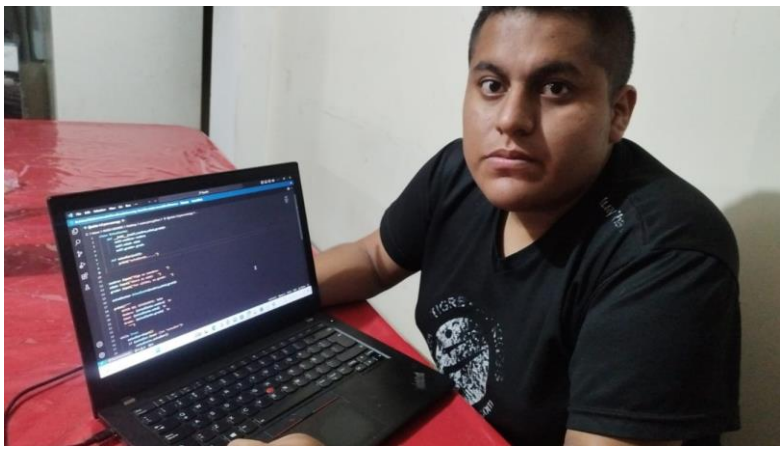
C: > Users > HUGO GRANDE > Desktop > curso poo python > HEREMCIA.py > Persona

```python
class Persona:
    def __init__(self,nombre,edad,nacionalidad):
        self.nombre= nombre
        self.edad=edad
        self.nacionalidad=nacionalidad


class Empleados(Persona):
    def __init__(self,nombre,edad,nacionalidad,trabajo, salario):
        super().__init__(nombre,edad,nacionalidad)
        self.trabajo=trabajo
        self.salario=salario

roberto= Empleados("Roberto",23,"ecuatoriano","programador",1000)

print(roberto.trabajo)
```



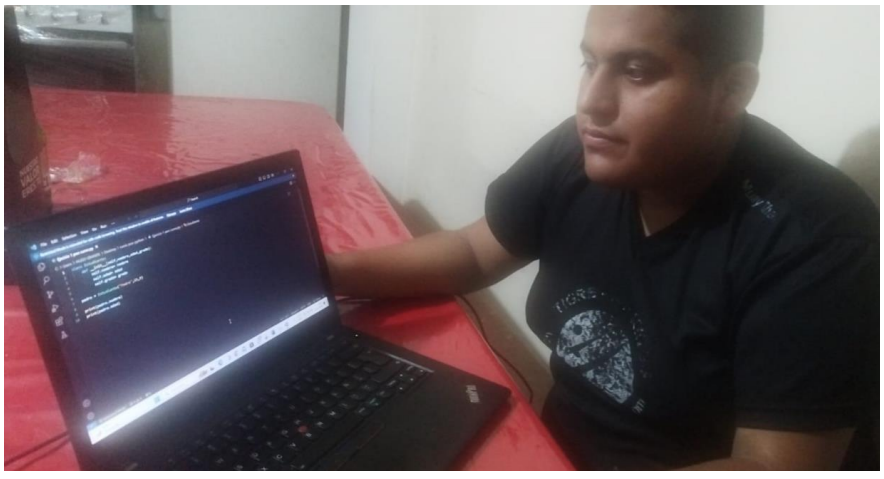
C: > Users > HUGO GRANDE > Desktop > curso poo python > Ejecicio 1.2 poo curso.py > ...

```python
class Estudiante:
    def __init__(self,nombre,edad,grado):
        self.nombre= nombre
        self.edad= edad
        self.grado= grado

    def estudiar(self):
        print("estudiando.....")



nombre= input("Diga su nomnbre:     ")
edad= input("Ahora su edad:   ")
grado= input("Por ultimo, su grado:   ")

estudiante= Estudiante(nombre,edad,grado)

print(f"""
    DATOS DEL ESTUDIANTE: \n\n
    Nombre: {estudiante.nombre}  \n
    Edad: {estudiante.edad}  \n
    Grado: {estudiante.grado}  \n
      """)

while True:
    estudiar=input()
    if (estudiar.lower ()== "estudiar"):
        estudiante.estudiar()
```



C: > Users > HUGO GRANDE > Desktop > curso poo python > Ejecicio 1 poo curso.py

```python
class Estudiante:
    def __init__(self,nombre,edad,grado):
        self.nombre= nombre
        self.edad= edad
        self.grado= grado

pedro = Estudiante("Pedro",23,9)

print(pedro.nombre)
print(pedro.edad)
```

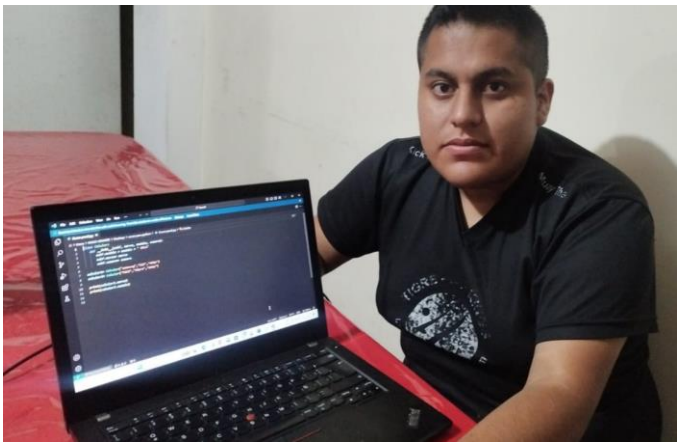


C: > Users > HUGO GRANDE > Desktop > curso poo python > Curso POO.py > Celular

```python
class Celular():
    marca = "iphone"
    modelo = "X"
    camara = "58MP"

celular1= Celular()
celular2= Celular()
celular3= Celular()
celular4= Celular()

print(celular4.marca)

```

C: > Users > HUGO GRANDE > Desktop > curso poo python > Curso poo2.py > Celular

```python
class Celular:
    def __init__(self, marca, modelo, camara):
        self.modelo = modelo + " HOLA"
        self.marca= marca
        self.camara= camara

celular1= Celular("samsung","S23","48Mp")
celular2= Celular("POCO","X5pro","68Mp")

print(celular2.marca)
print(celular2.modelo)
```



C: > Users > HUGO GRANDE > Desktop > curso poo python > Curso poo3.py > Celular

```python
class Celular:
    def __init__(self, marca, modelo, camara):
        self.modelo = modelo
        self.marca= marca
        self.camara= camara


    def llamar(self):
        print(f"Estas haciendo una llamada a tu amigo desde un: {self.modelo}")


    def cortar(self):
        print(f"Cortastes la llamada desde tu: {self.marca}")


celular1= Celular("samsung","S23","48Mp")
celular2= Celular("POCO","X5pro","68Mp")

celular1.cortar()
celular2.llamar()

```