

**TALLER No. 3 PYTHON**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

El método **isinstance**, sirve para saber si una instancia pertenece a una clase, este método recibe dos argumentos la instancia y la clase, retorna True o False si pertenece o no a la clase.

**pass** es una instrucción importante para Python ya que permite dejar “pasar” la ejecución y no generar error en métodos o clases sin cuerpo.

1. La siguiente es la estructura del paquete taller:

taller/

+-- src/

+-- main.py

+-- almacenamientos/

+-- almacenamiento.py

+-- ram.py

+-- dispositivos/

+-- init .py

+-- almacenamientoExterno.py

+-- celular.py

+-- computador.py

¿Es correcta esta estructura? ¿Si le falta algo, que sería?

1. Analice el siguiente ejercicio y cambie los lugares con \*\*, con el valor que debería ir, para que funcione, cumpla con el encapsulamiento y tenga un uso correcto de los módulos.

**Nota**: la estructura de los archivos es la especificada en el punto **a**

**Resultado esperado al ejecutar main.py**

Celular Realme 8 pro Cantidad de celulares: 1

Cantidad de almacenamientos externos: 1 Cantidad de computadores: 1 ['\_Computador cantidad', ' class ',

' delattr ', ' dict ', ' dir ', ' doc ',

' eq ', ' format ', ' ge ',

' getattribute ', ' gt ', ' hash ', ' init ', ' init\_subclass ', ' le ',

' lt ', ' module ', ' ne ', ' new ', ' reduce ', ' reduce\_ex ', ' repr ',

' setattr ', ' sizeof ',

' str ', ' subclasshook ', ' weakref ', 'getConexion', 'getMarca', 'getModelo', 'setConexion', 'setMarca']