



## **SOLUCIÓN TALLER # 3 (PYTHON/JAVA)**

### **ASIGNATURA**

Programación Orientada a Objetos

### **PROFESOR**

[Jaime Alberto Guzmán Luna](#)

### **ESTUDIANTE**

**SAMUEL TORO AGUDELO**

### **NÚMERO DE IDENTIFICACIÓN**

**1020107389**

**Universidad Nacional de Colombia**

**Sede Medellín**

**2024**



## RESPUESTA PREGUNTAS

A) Esta bien la estructura del paquete

B) Archivo main.py

```
from dispositivos.celular import Celular
from dispositivos.almacenamientoExterno import AlmacenamientoExterno
from dispositivos.computador import Computador
from almacenamientos.ram import Ram
from almacenamientos.almacenamiento import Almacenamiento

if __name__ == "__main__":
    cel = Celular("Alejandro", "Realme 8 pro", Almacenamiento(128, "SSD"))
    alm = AlmacenamientoExterno("Curso POO")
    pc = Computador("Lenovo", "s330", Almacenamiento(2000, "HDD"), Ram(8,
"DDR4"))

    pc.setConexion(cel)

    print(pc.getConexion())
    print("Cantidad de celulares: " + str(Celular.getCantidad()))
    print("Cantidad de almacenamientos externos: " +
str(AlmacenamientoExterno.getCantidad()))
    print("Cantidad de computadores: " + str(Computador.getCantidad()))
    print(dir(pc))
```

Archivo almacenamiento.py

```
class Almacenamiento:
    def __init__(self, tamaño, tipo):
        self._TAMAÑO = tamaño
        self.tipo = tipo

    def getTamano(self):
        return self._TAMAÑO

    def setTamaño(self, tamaño):
        self._TAMAÑO = tamaño
```

Archivo ram.py

```
class Ram:
    def __init__(self, tamaño, tipoModulo):
```



```
self._TAMAÑO = tamaño
self._tipoModulo = tipoModulo
```

```
def setTipoModulo(self, tipoModulo):
    self._tipoModulo = tipoModulo
```

```
def getTipoModulo(self):
    return self._tipoModulo
```

```
def getTamaño(self):
    return self._TAMAÑO
```

Archivo almacenamientoExterno.py

```
from almacenamientos.almacenamiento import Almacenamiento
```

```
class AlmacenamientoExterno:
    _cantidad = 0

    def __init__(self, nombre):
        self._nombre = nombre
        self._almacenamiento = Almacenamiento(1000, "HDD")
        AlmacenamientoExterno._cantidad += 1

    def setNombre(self, nombre):
        self._nombre = nombre

    def getNombre(self):
        return self._nombre

    def setAlmacenamiento(self, almacenamiento):
        self._almacenamiento = almacenamiento

    def getAlmacenamiento(self):
        return self._almacenamiento

    @classmethod
    def getCantidad(cls):
        return cls._cantidad
```

Archivo celular.py

```
class Celular:
    _cantidad = 0
```



```
def __init__(self, dueno, modelo, almacenamiento):  
    self.dueno = dueno  
    self.modelo = modelo  
    self.almacenamiento = almacenamiento  
    Celular._cantidad += 1
```

```
@classmethod  
def getCantidad(cls):  
    return cls._cantidad
```

Archivo computador.py

```
from dispositivos.celular import Celular
```

```
from dispositivos.almacenamientoExterno import AlmacenamientoExterno
```

```
class Computador:
```

```
    _cantidad = 0
```

```
def __init__(self, marca, modelo, almacenamiento, ram):  
    self._marca = marca  
    self._modelo = modelo  
    self.almacenamiento = almacenamiento  
    self.ram = ram  
    self._conectado = None  
    Computador._cantidad += 1
```

```
def setMarca(self, marca):  
    self._marca = marca
```

```
def getMarca(self):  
    return self._marca
```

```
def getModelo(self):  
    return self._modelo
```

```
def setConexion(self, dispositivo):  
    self._conectado = dispositivo
```

```
def getConexion(self):  
    if(isinstance(self._conectado, Celular)):
```



```
        return "Celular " + self._conectado.modelo
    elif(isinstance(self._conectado, AlmacenamientoExterno)):
        return "Almacenamiento " + self._conectado.getNombre()
```