



Vamos a suponer que tenemos una tienda de ensamblaje de dispositivos (PC's de mesa, Portátiles, Tablets y Móviles).

Estos son dispositivos que se pueden ensamblar a gusto del cliente, cada dispositivo tiene un precio base y cada componente se acumula al precio base.

1. Se creará una clase **CPU** que tendrá como atributos:

- int velocidad en megahertzios , int numeroNucleos, String modelo y int precio.

Esta clase tendrá un constructor al que le podremos pasar los siguientes parámetros (String modelo, int velocidad, int numeroNucleos, int precio).

Sus setters / getters y un método toString que nos dará la siguiente salida por pantalla:

```
CPU
Velocidad=5000
NumeroNucleos=10
Modelo=i9-10910
Precio=500
```

2. Se creará una clase **RAM** que tendrá como atributos:

- int capacidad en gigas, String modelo, int precio

Tendrá un constructor al que le podremos pasar los parámetros (String modelo, int capacidad, int precio)

Sus setters / getters y un método toString que nos dará la siguiente salida:

```
RAM
Capacidad=16
Modelo=Crucial Ballistix BL2K8G36C16U4R
Precio=112
```

3. Se creará una clase **HDD** que tendrá como atributos:

- int capacidad en gigas, String tipo (sata, sata2, ssd, MV2 ...), int precio;

Tendrá un constructor al que le podremos pasar como parámetros (String tipo, int capacidad, int precio)

Sus setters / getters y un método toString que nos dará la siguiente salida:

```
HDD
Capacidad=500
Tipo=SATA2
Precio=55
```

4. Se creará una clase **Bateria** que tendrá como atributos:

- String modelo, int capacidad en miliamperios, int nivelBateria (porcentaje), int precio.

Tendrá un constructor al que le podremos pasar como parámetros (String modelo, int capacidad, int nivelBateria, int precio)

Sus setters / getters y un método toString que nos dará la siguiente salida:

```
BATERIA
```



Capacidad=3500
NivelBateria=100
Modelo=Green Cell® L16M2PB1 L16C2PB2
Precio=40

5. Se creará la clase **Dispositivo**, la cual no podremos instanciar, y que tendrá como atributos:

- cpu de la clase CPU, ram de la clase RAM, precioBase que será un entero, discos que será un arrayList de HDD.

Definiremos los setters y getters

Un método addDisk ara poder añadir un HDD a nuestro arrayList de HDD

Un método getDiskCapacity que nos devolverá la capacidad total de todos los discos de nuestro arrayList de HDD.

6. Se creará la interface **Recargable**, que tendrá un único método recargarBateria()

7. Se creará la clase **Tablet**, que tendrá los mismos atributos que la clase Dispositivo , un atributo bateria de la clase Bateria, y además implementará la interfaz Recargable.

Tendrá un constructor al cual le podremos pasar los siguientes parámetros (CPU cpu, RAM ram, Bateria bateria, HDD disco, int precioBase)

Los setters / getters que sean necesarios para el ejercicio.

Un método getPrecio que nos calcule el precio total de la tablet, en función del precio base y de los componentes que tenga.

Un método toString que nos muestre por pantalla lo siguiente:

Tablet
Cpu=CPU
Velocidad=2000
NumeroNucleos=8
Modelo=MediaTek Dimensity 1000+
Precio=130

RAM
Capacidad=8
Modelo=LPDDR4X
Precio=50

Discos

HDD
Capacidad=256
Tipo=UFS 3.1
Precio=90
Capacidad total almacenamiento=256

BATERIA
Capacidad=5000
NivelBateria=100



Modelo=Li-Polymer

Precio=65

Precio 425

8. Se creará la clase **Movil**, que tendrá los mismos atributos que la clase *Dispositivo* , un atributo bateria de la clase Bateria, y además implementará el la interfaz Recargable.

Tendrá un constructor al cual le podremos pasar los siguientes parámetros (CPU cpu, RAM ram, Bateria bateria, HDD disco, int precioBase)

Los setters / getters que sean necesarios para el ejercicio.

Un método getPrecio que nos calcule el precio total del movil, en función del precio base y de los componentes que tenga.

Un método toString que nos muestre por pantalla lo siguiente:

Movil

CPU

Velocidad=1800

NumeroNucleos=8

Modelo=Qualcomm Snapdragon 865

Precio=120

RAM

Capacidad=6

Modelo=LPDDR5

Precio=40

Discos

HDD

Capacidad=128

Tipo=UFS 3.1

Precio=60

Capacidad total almacenamiento=128

BATERIA

Capacidad=4250

NivelBateria=100

Modelo=Li-Polymer

Precio=55

Precio 325

9. Se creará la clase **Portatil**, que tendrá los mismos atributos que la clase *Dispositivo* , un atributo bateria de la clase Bateria, y además implementará el la interfaz Recargable.

Tendrá un constructor al cual le podremos pasar los siguientes parámetros (CPU cpu, RAM ram, Bateria bateria, HDD disco, int precioBase)



Los setters / getters que sean necesarios para el ejercicio.

Un método getPrecio que nos calcule el precio total del portatil, en función del precio base y de los componentes que tenga.

Un método toString que nos muestre por pantalla lo siguiente:

Portatil

CPU

Velocidad=3800

NumeroNucleos=4

Modelo=i7-4900MQ

Precio=270

RAM

Capacidad=8

Modelo=Viper Elite DDR4 2666

Precio=57

Discos

HDD

Capacidad=250

Tipo=SSD

Precio=110

Capacidad total almacenamiento=250

BATERIA

Capacidad=3500

NivelBateria=100

Modelo=Green Cell® L16M2PB1 L16C2PB2

Precio=40

Precio 627

10. Se creará la clase PCMesa, que tendrá los mismos atributos que la clase Dispositivo.

Tendrá un constructor al cual le podremos pasar los siguientes parámetros (CPU cpu, RAM ram, HDD disco, int precioBase)

Los setters / getters que sean necesarios para el ejercicio.

Un método getPrecio que nos calcule el precio total del PC, en función del precio base y de los componentes que tenga.

Un método toString que nos muestre por pantalla lo siguiente:

PCMesa

Cpu=CPU

Velocidad=5000

NumeroNucleos=10



Modelo=i9-10910

Precio=500

RAM

Capacidad=16

Modelo=Crucial Ballistix BL2K8G36C16U4R

Precio=112

Discos

HDD

Capacidad=500

Tipo=SATA2

Precio=55

Capacidad total almacenamiento=500

Precio 667

11. Crearemos una clase principal donde crearemos al menos un dispositivo de cada clase, Tablet, Movil, Portatil, PCMesa.

Como necesitaremos crear los componentes queramos ensamblar, se dejan los siguientes a modo de ejemplo

```
CPU cpu1 = new CPU("i9-10910",5000,10, 500);
```

```
CPU cpu2 = new CPU("i7-4900MQ",3800,4, 270);
```

```
CPU cpu3 = new CPU("Qualcomm Snapdragon 865",1800,8, 120);
```

```
CPU cpu4 = new CPU("MediaTek Dimensity 1000+",2000,8, 130);
```

```
RAM ram1 = new RAM("Crucial Ballistix BL2K8G36C16U4R", 16, 112);
```

```
RAM ram2 = new RAM("Viper Elite DDR4 2666", 8, 57);
```

```
RAM ram3 = new RAM("LPDDR5", 6, 40);
```

```
RAM ram4 = new RAM("LPDDR4X", 8, 50);
```

```
HDD hdd1 = new HDD("SATA2", 500, 55);
```

```
HDD hdd2 = new HDD("SSD", 250, 110);
```

```
HDD hdd3 = new HDD("UFS 3.1", 128, 60);
```

```
HDD hdd4 = new HDD("UFS 3.1", 256, 90);
```

```
Bateria b2 = new Bateria("Green Cell® L16M2PB1 L16C2PB2 ", 3500, 100,40);
```

```
Bateria b3 = new Bateria("Li-Polymer", 4250, 100,55);
```

```
Bateria b4 = new Bateria("Li-Polymer", 5000, 100,65);
```

Se creará un arrayList al que llamaremos **carritoCompra** donde iremos añadiendo los dispositivos creados.



Crearemos un método que sea mostrar el carrito de la compra, que nos muestre por pantalla todos los dispositivos que componen el carrito de la compra (mostraremos la salida dispositivo a dispositivo, no utilizaremos un simple `System.out.println(carritoCompra)`);

Crearemos un método que nos muestre los dispositivos ordenados por precio de forma ascendente, para ello nos haremos uso de la interfaz **Comparable**.

Crearemos un método que nos muestre los dispositivos ordenados por la capacidad total de almacenamiento de forma ascendente, para ello haremos uso de la interfaz **Comparator**.