

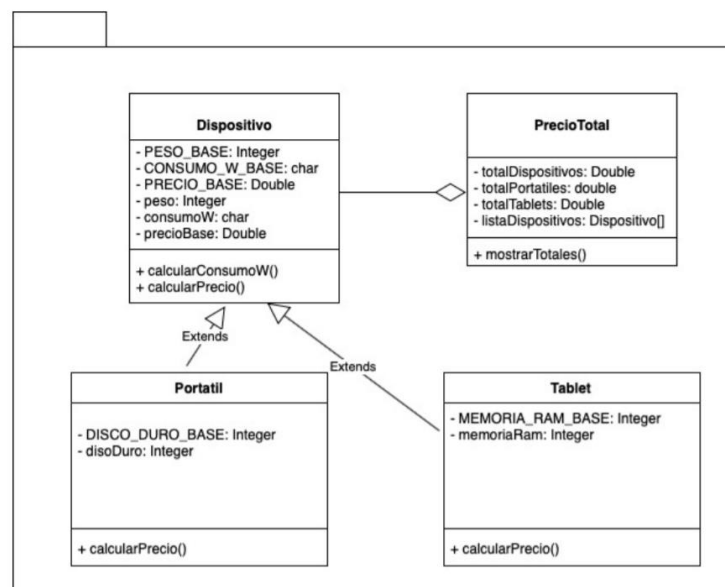


## Ciclo 2 Fundamentos de programación

### Reto 2

#### Descripción del problema:

La UNICEF, en su afán de llevar calidad de vida a todos los niños del mundo, en especial a países con alto riesgo, ha determinado un sistema de donaciones para favorecer a los más desamparados. En consecuencia, se ha abierto una convocatoria para recibir donaciones de computadores y tabletas en la búsqueda de poder llevar a todos los lugares posibles, educación, pese al distanciamiento social producto de la crisis pandémica. Es por ello, que, como una labor adicional al proceso es necesario determinar el valor total por cada tipo de dispositivo electrónico, así como el valor total para todos los dispositivos electrónicos donados. Para realizar esta actividad, se ha contratado su compañía, en donde se ha determinado que el modelo de clases mediante el cual se resolverá el problema es el siguiente:



Todos los dispositivos electrónicos comparten los atributos peso y consumoW (consumo de vatios), los cuales son valores que pueden modificar su precio final, se cuenta también con el atributo precioBase, el cual representa el precio del dispositivo electrónico antes de sumar los respectivos valores según la evaluación de los atributos peso y consumoW. El método calcularPrecio() permite obtener el precio final para un cualquier dispositivo electrónico de la siguiente forma:

$$\text{precioFinal} = \text{precioBase} + \text{adición}$$

La adición se establece según los siguientes criterios:

Con respecto al consumo energético; si éste es mayor a 10W, se toma el valor recibido, de lo contrario, se toma el valor por defecto de la constante CONSUMO\_W.

Por otra parte, si el consumo energético (consumoW) del dispositivo es 'A', se adiciona \$100, si es 'B' se adiciona \$80, si es 'C' se adiciona \$60, si es 'D' se adiciona \$50, si es 'E' se adiciona \$30 y si es 'F' se adiciona \$10.



Asimismo, si el peso del dispositivo es mayor a 1 y menor o igual a 2 se adiciona \$10, si el peso es mayor a 2 y menor o igual a 3 se adiciona \$50, si el peso es mayor a 3 y menor o igual a 4 se adiciona \$80 y si el peso es mayor a 4s se adiciona \$100.

Por otra parte, el dispositivo electrónico de tipo “Portátil” (Computador portátil), cuentan con el atributo discoDuro, el cual agrega un valor adicional al precio del ítem de la siguiente forma:

Si el discoDuro es mayor a 250 y menor o igual a 500 Gb, se adiciona \$10.

Si el discoDuro es superior a 500 Gb y menor o igual a 1000 Gb, se adiciona \$50.

Si el discoDuro es superior a 1000 Gb, se adiciona \$100.

**Importante:** Por defecto, los computadores portátiles tienen 250 Gb de disco duro.

DD\_BASE = 250 (Gb)

Si no se cuenta con valores para estos atributos, se deben inicializar con estos valores constantes.

En otro respecto, concerniente a las tabletas, éstas cuentan con el atributo memoriaRam, la cual adiciona un valor extra al precio del ítem de la siguiente forma:

Si la memoriaRam es superior a 1 Gb y menor o igual a 2 Gb, se adiciona \$5.

Si la memoriaRam es superior a 2 Gb y menor o igual a 4 Gb, se adiciona \$25.

Si la memoriaRam es superior a 4 Gb, se adiciona \$50.

Por defecto, las tabletas tienen 1 Gb de memoria RAM, así:

MEMORIA\_RAM\_BASE = 1 (Gb)

Si no se cuenta con valores para este atributo, se deben inicializar con estos valores constantes.

Cada uno de los atributos de la clase dispositivos electrónicos, tienen una constante que define un valor por defecto, con el cual, se puede calcular el precio del dispositivo si no se envía ningún valor al constructor de la clase.

CONSUMO\_W\_BASE = 'F'

PRECIO\_BASE = 100.0

PESO\_BASE = 1

Si no se cuenta con valores para estos atributos, se deben inicializar con estos valores constantes.

Los atributos de la clase PrecioTotal son: totalDispositivos, totalPortatiles, totalTablets y listaDispositivos. Por Defecto, sus valores son 0. Este último atributo contiene todos los dispositivos electrónicos recibidos en donación, los cuales son almacenados en un array (tipo Dispositivo), estos son instanciados en la clase App en su método main(), para ser entregados al constructor de la clase PrecioFinal.

Dentro del método main(), se llama además al método mostrarTotales(), el cual debe imprimir en consola:

Total precios de dispositivos {totalDispositivos}

Total precios de computadores portátiles { totalPortatiles }

Total precios de tabletas { totalTablets }



### Ejemplo:

Dispositivo dispositivos[]=new Dispositivo[3]; dispositivos[0]=new Portatil(500.0, 3, 'E', 250); dispositivos[1]=new Tablet(); dispositivos[2]=new Dispositivo(600.0, 3, 'D'); PrecioTotal solucion1 = new PrecioTotal(dispositivos); solucion1.mostrarTotales();	Total precios de dispositivos 1330.0 Total precios de computadores portátiles 560.0 Total precios de tabletas 110.0
Dispositivo dispositivos2[]=new Dispositivo[7]; dispositivos2[0]=new Tablet(150.0, 1); dispositivos2[1]=new Portatil(500.0, 2, 'E', 500); dispositivos2[2]=new Dispositivo(); dispositivos2[3]=new Portatil(250.0, 4); dispositivos2[4]=new Tablet(400.0, 3, 'A', 4); dispositivos2[5]=new Dispositivo(50.0, 3); PrecioTotal solucion2 = new PrecioTotal(dispositivos2); solucion2.mostrarTotales();	Total precios de dispositivos 1735.0 Total precios de computadores portátiles 870.0 Total precios de tabletas 645.0

### IMPORTANTE

Para trabajar con el siguiente esquema su archivo .java debe nombrarse App.java

Las pruebas son ejecutadas en la clase App. Esta clase no se debe subir a la plataforma como parte de la solución.

### Esqueleto:

```
// Inicio de la solución
public class PrecioTotal {
    // Atributos

    // Constructores
    PrecioTotal(Dispositivo[] pDispositivos) {

    }

    // Metodos
    public void mostrarTotales() {
        // Calculos de totales

        System.out.println("Total precios de dispositivos " + totalDispositivos);
        System.out.println("Total precios de computadores portátiles " + totalPortatiles);
        System.out.println("Total precios de tabletas " + totalTablets);
    }
}

class Dispositivo {
    // Constantes y atributos

    //Constructores
    public Dispositivo(){

    }

    public Dispositivo(Double precioBase, Integer peso){
        //codigo
    }

    public Dispositivo(Double precioBase, Integer peso, char consumoW){
        //codigo

        comprobarconsumoW(consumoW);
    }

    // Metodos
    public void comprobarconsumoW(char consumoW){

        if(//condicion){
            this.consumoW=consumoW;
        }else{
            this.consumoW=CONSUMO_W;
        }
    }
}
```



```
        public Double calcularPrecio(){
            //codigo

            return precioBase+adicion;
        }
    }

class Tablet extends Dispositivo{

    //Constantes y atributos

    //Constructor
    public Tablet(){

        //codigo

    }

    public Tablet(Double precioBase, Integer peso){

    }

    public Tablet(Double precioBase, Integer peso, char consumoW, Integer memoriaRam){

        //codigo

    }

    // Métodos
    public Double calcularPrecio(){
        //codigo

    }
}

class Portatil extends Dispositivo{

    //Constantes y atributos

    //Constructor
    public Portatil(){

        //codigo

    }

    public Portatil(Double precioBase, Integer peso){
        //codigo

    }

    public Portatil(Double precioBase, Integer peso, char consumoW, Integer discoDuro){

        //codigo

    }

    // Métodos
    public Double calcularPrecio(){
        //codigo

    }
}

// Fin de la solución

// Esta clase es para las pruebas, no se debe subir como parte de la solución
public class App {
    public static void main(String[] args) {
        // Pruebas Públicas
        // Pruebas Publicas
        Dispositivo dispositivos[]=new Dispositivo[3];
        dispositivos[0]=new Portatil(500.0, 3, 'E', 250);
        dispositivos[1]=new Tablet();
        dispositivos[2]=new Dispositivo(600.0, 3, 'D');
        PrecioTotal solucion1 = new PrecioTotal(dispositivos);
        solucion1.mostrarTotales();
        System.out.println("\n");

    }
}
```