

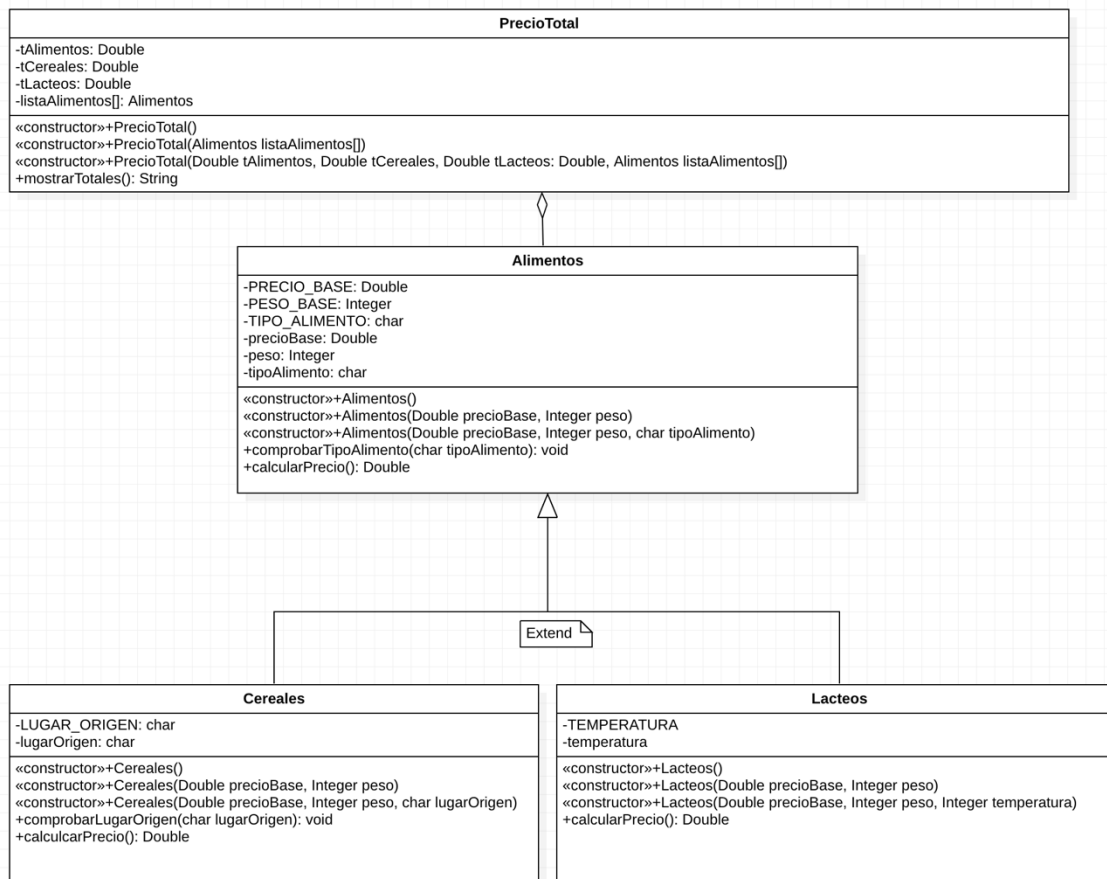


## Ciclo 2 Fundamentos de programación

### Reto 2

#### Descripción del problema:

Una nueva comercializadora de alimentos desea adquirir nuevos productos de los cuales necesita determinar el valor total por cada uno de ellos y total de los productos alimentos. Para lo cual se contrata sus servicios de desarrollador, en donde se ha determinado el modelo de clases que nos permite gestionar el programa el cual es el siguiente:



Todos los productos alimentos comparten los atributos *pesoBase*, *tipoAlimento*, los cuales son valores que pueden modificar su precio final, se cuenta también con el atributo *precioBase*, el cual representa el precio del productos alimentos antes de sumar los respectivos valores según la evaluación de los atributos *pesoBase*, *tipoAlimento*. El método *calcularPrecio()* permite obtener el precio final para cualquier producto alimento de la siguiente forma:

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

La adición se establece según lo siguiente:

Si el tipo de alimento(*tipoAlimento*) del producto alimentos es 'N' (Natural), se le adiciona \$40, 'C' (Conservantes) se adiciona \$20.



Del mismo modo, si el peso del producto alimento es mayor o igual a 0 y menor o igual a 9 se adiciona \$6, si el peso es mayor a 9 y menor o igual a 16 se le adiciona \$8, si es mayor a 16 se adiciona \$20.

Los productos alimentos del tipo cereales cuentan con el atributo `lugarOrigen`, el cual agrega un valor adicional al precio de la cereales de la siguiente forma:

Si la `lugarOrigen` de la cereales es 'N' (Nacional), se adiciona \$5, si es 'I' (Internacional) se adiciona \$15.

`LUGAR_ORIGEN = 'N'`

Si no se tiene valor para el atributo `lugarOrigen`, este debe inicializarse con el valor de la contante.

Los productos alimentos del tipo lacteos cuentan el atributo `temperatura`, el cual agrega un valor adicional al precio de los lacteos de la siguiente forma:

Si la temperatura del lacteos es mayor o igual 5 grados y menor 10 grados se adiciona \$20, si la temperatura es mayor a igual 10 y menor o igual a 15 se le adiciona \$18, si es menor a 5 o mayor a 15 se adiciona \$5.

Se tiene un valor constante para la temperatura

`TEMPERATURA = 10`

Si no se tiene valor para el atributo `temperatura`, este debe inicializarse con el valor de la contante.

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

`TIPO_ALIMENTO = 'N'`

`PRECIO_BASE = 80.0`

`PESO_BASE = 5`

Si no se tiene valor para los atributos, este debe inicializarse con el valor de la contante

Los atributos de la clase `PrecioTotal` son: `tAlimentos`, `tCereales`, `tLacteos` y `listaAlimentos`. Este último atributo contiene todos los productos alimentos recibidos, los cuales son almacenados en un array (tipo producto alimento) y son entregados al constructor de la clase `PrecioTotal`.

En el método `main()`, desde donde se llama al método `mostrarTotales()`, el cual debe imprimir en consola:

El precio total de los Productos Alimentos es de { `tAlimentos` }

La suma del precio de los Lácteos es de { `tCereales` }

La suma del precio de los Cereales es de { `tLacteos` }



### Ejemplo:

Pruebas	Salida
<pre>Alimentos listaAlimentos1[] = new Alimentos[5];     listaAlimentos1[0] = new Alimentos();     listaAlimentos1[1] = new Cereales(204.1,18,'N');     listaAlimentos1[2] = new Lacteos(120.5,40);     listaAlimentos1[3] = new Cereales(180.0,60,'I');     listaAlimentos1[4] = new Lacteos(190.3,20);  PrecioTotal solucion1 = new PrecioTotal(listaAlimentos1); solucion1.mostrarTotal();</pre>	<p>El precio total de los Productos Alimentos es de 1116.9</p> <p>La suma del precio de los Cereales es de 524.1</p> <p>La suma del precio de los Lácteos es de 466.8</p>
<pre>Alimentos listaAlimentos2[] = new Alimentos[10];     listaAlimentos2[0] = new Alimentos(62.8,56,'N');     listaAlimentos2[1] = new Alimentos(23.5,16,'I');     listaAlimentos2[2] = new Cereales(212.0,90,'N');     listaAlimentos2[3] = new Lacteos(60.6,90,2);     listaAlimentos2[4] = new Cereales(60.0,6);     listaAlimentos2[5] = new Lacteos();     listaAlimentos2[6] = new Lacteos(87.2,240,11);     listaAlimentos2[7] = new Cereales(343.1,140,'N');     listaAlimentos2[8] = new Lacteos();     listaAlimentos2[9] = new Lacteos(80.0,40,1);  PrecioTotal solucion2 = new PrecioTotal(listaAlimentos2); solucion2.mostrarTotal();</pre>	<p>El precio total de los Productos Alimentos es de 1714.1999999999998</p> <p>La suma del precio de los Cereales es de 796.1</p> <p>La suma del precio de los Lácteos es de 723.8</p>

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

### Esqueleto:

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.

```
public class PrecioTotal {  
  
    // Atributos  
  
    // Constructores
```



```
public PrecioTotal(){  
  
}  
  
public PrecioTotal(Alimentos listaAlimentos[]){  
}  
  
// Metodo mostrar Total  
  
public void mostrarTotal(){  
  
    // Codigo calculo de total  
  
    System.out.println("El precio total de los Productos Alimentos es de " +  
tAlimentos);  
    System.out.println("La suma del precio de las Cereales es de " + tCereales);  
    System.out.println("La suma del precio de los Lácteos es de " + tLacteos);  
  
}  
}  
  
public class Alimentos {  
  
    // Atributos  
  
    // Constructores  
    public Alimentos(){  
  
    }  
  
    public Alimentos(Double precioBase, Integer peso){  
  
    }  
  
    public Alimentos(Double precioBase, Integer peso, char tipoAlimento){  
  
    }  
  
    public void comprobarTipoAlimento(char tipoAlimento){  
        // Condicion validar tipo de alimento N,C  
    }  
  
    public Double calcularPrecio(){  
        // Codigo calcular precio final  
  
        return precioFinal;  
    }  
}
```



```
public class Cereales extends Alimentos {

    // Atributos

    // Constructores

    public Cereales(){

    }

    public Cereales(Double precioBase, Integer peso){

    }

    public Cereales(Double precioBase, Integer peso, char lugarOrigen){

    }

    public void comprobarLugarOrigen(char lugarOrigen){
        // Condicion validar lugar de origen N,I
    }

    public Double calcularPrecio(){
        // Codigo calcular precio final
        return precioFinal;
    }

}

public class Lacteos extends Alimentos {

    // Atributos

    // Constructores
    public Lacteos(){

    }

    public Lacteos(Double precioBase, Integer peso){

    }

    public Lacteos(Double precioBase, Integer peso, Integer temperatura){

    }

    public Double calcularPrecio(){
```



```
// Codigo calcular precio final
return precioFinal;
}

}

// 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 )
    {
        Alimentos listaAlimentos1[] = new Alimentos[5];
        listaAlimentos1[0] = new Alimentos();
        listaAlimentos1[1] = new Cereales(204.1,18,'N');
        listaAlimentos1[2] = new Lacteos(120.5,40);
        listaAlimentos1[3] = new Cereales(180.0,60,'I');
        listaAlimentos1[4] = new Lacteos(190.3,20);

        PrecioTotal solucion1 = new PrecioTotal(listaAlimentos1);
        solucion1.mostrarTotal();
        System.out.println();
    }
}
```