

Examen Parcial 2 - Laboratorio de Computación 1

Dados las siguientes variables:

```
String [][] tiposComprobante = {"FV","Factura Venta"}, {"R","Remito"}, {"P","Presupuesto"}, {"PV","Pedido Venta"}, {"FO","Factura Compra"};
```

```
HashMap<Integer, String> articulos = new HashMap<>();
    articulos.put(100,"Arroz");
    articulos.put(200,"Lavandina");
    articulos.put(300,"Fideos");
    articulos.put(400,"Leche");
    articulos.put(500,"Harina");
    articulos.put(600,"Aceite");
    articulos.put(700,"Sal");
    articulos.put(800,"Pimienta");
    articulos.put(900,"Detergente");
    articulos.put(1000,"Yerba");
    articulos.put(1100,"Shampoo");
    articulos.put(1200,"Jabon");
    articulos.put(1300,"Cepillo de Dientes");
    articulos.put(1400,"Pasta Dental");
    articulos.put(1500,"Lentejas");
```

Codifique las siguientes clases Java

Clase: Comprobante

Atributos: nroComprobante (Long), fecha (String), tipoComprobante (String), itemsComprobante (ArrayList<DetalleComprobante>)

Clase: DetalleComprobante

Atributos: articulo (String), cantidad (double), precio (double)

Clase: AltaComprobante (clase que posee el método main)

Lógica a Implementar:

- 1- En la clase **AltaComprobante** cree una instancia de la clase **Comprobante** y solicite inicialmente los datos del mismo, validando que numero de comprobante, fecha y tipo de comprobante no sean vacío. El tipo de comprobante deberá obtenerse del array de 2 dimensiones **tiposComprobante**, solicitando al usuario que indique el identificador del mismo.
- 2- Posteriormente comenzara la creación y carga de los ítems del comprobante creado en el paso anterior, validando que articulo, cantidad y precio no sean vacío y que cantidad y precio sean mayor a 0. El articulo deberá asignarse solicitando al usuario que ingrese el código del mismo (si el código no existe indicarlo por pantalla), y obtener el nombre correspondiente del HashMap.

- El paso número 2 se repetirá hasta que el usuario indique que no quiere continuar cargando ítems al comprobante.
- Valide que al menos se agregue 1 ítem al comprobante.

- 3- Agregue a la clase **DetalleComprobante** el método:

public double calcularSubTotal (){}

que retorne la multiplicación del precio por la cantidad.

- 4- Agregue a la clase **Comprobante** el método:

public double calcularTotalComprobante(){}

que retorne la suma de los subtotales de los ítems del comprobante calculados en el paso anterior. Reutilice el método del punto 3.

- 5- Finalmente mostrar por pantalla la siguiente información:

Comprobante Tipo: XXXXXXXX

N° Comprobante: XXXXXXXXXXXX Fecha: XXXXXXXXXX

----- **Items del Comprobante** -----

Articulo	Cantidad	Precio Unidad	Subtotal
xxxxx	xxxxx	xxxxx	xxxxx

Total Comprobante: xxxxx