

## TP - Colecciones con Asociación en JAVA

1. **Modifique el examen parcial 1**, eliminando el array de 2 dimensiones **detallesFertilizantes** reemplazando el mismo por un ArrayList <DDJJFertilizanteIngresoDetalle> de objetos DDJJFertilizanteIngresoDetalle.

La clase DDJJFertilizanteIngresoDetalle estará compuesta por los atributos:

Código Fertilizante (String)  
Nombre Fertilizante (String)  
Tipo Envase (String)  
Capacidad Envase (double)  
Cantidad Envases (int)  
Subtotal (double)

La aplicación debe solicitar y almacenar en el ArrayList tantos agroquímicos como se desee, al finalizar mostrar la información respetando los lineamientos del parcial.

2. **Cree la clase java Factura la cual debe contener los siguientes atributos privados:**

- **fechaFactura** de tipo String
- **numeroFactura** de tipo long
- **totalCalculadoFactura** de tipo double
- **cliente** de tipo String
- **detallesFactura** ArrayList de objetos **DetalleFactura**

**La clase DetalleFactura debe contener los siguientes atributos:**

Código Artículo (String)  
Nombre Artículo (String)  
Cantidad (int)  
Precio Unitario (double)  
Descuento Ítem (double)  
Subtotal (double)

Genere los correspondientes métodos get / set públicos para cada una de las variables de la clase

Cree la clase java **Facturación** que contenga la definición de una variable "**artículos**" que se corresponde al siguiente **array** de 2 dimensiones, 10 filas por 3 columnas, definido como variable global en la clase con los siguientes artículos cargados:

101	Leche	25
102	Gaseosa	30
103	Fideos	15
104	Arroz	28
105	Vino	120
106	Manteca	20
107	Lavandina	18
108	Detergente	46
109	Jabón en Polvo	96
110	Galletas	60

Las columnas se corresponden con código de artículo, nombre del artículo y precio del artículo.

Codifique en la clase **Facturación** un método main inicial que contenga el código necesario para ejecutar las siguientes acciones:

- a. Crear una instancia (**new**) de la clase Factura
- b. Solicitar los datos de fecha factura, número de factura y cliente y asignarlos en la instancia de factura creada en el punto anterior. Validaciones número de factura debe ser numero entero mayor a cero y cliente debe ser distinto de vacío
- c. Solicitar el código del articulo a facturar, el usuario ingresa un valor, buscar el código en el array de artículos, si no se encuentra el código indicar la situación con el mensaje "El código ingresado no existe, intente nuevamente" y volver a pedir el código, si el código se encuentra solicitar la cantidad a facturar y aplicar la información obtenida para cargar el Objeto DetalleFactura, finalizada la carga del objeto agregar el mismo en el ArrayList de **detallesFactura** de la instancia, si la cantidad ingresada es mayor a 5 se aplica un 10% de descuento sobre el precio del artículo, el subtotal es el resultante de multiplicar precio unitario de articulo por cantidad (con o sin descuento según corresponda). **Repetir** este proceso hasta que el usuario indique que no quiere continuar cargando Articulos a Facturar en el ArrayList **detallesFactura**.

**Ejemplo:** si ingreso "104" e indicó en cantidad el valor 7, el objeto DetalleFactura debe cargarse con los datos:

Código Artículo	Nombre Artículo	Cantidad	Precio Unitario	Descuento Ítem	Subtotal
104	Arroz	7	28	2.8	176.4

Siendo: **Monto descuento** =  $28 * 0.1 = 2.8$

**Subtotal** =  $(28 * 7) - (2.8 * 7) = 196 - 19.6 = 176.4$

- d. Codifique en la clase Factura un método:

```
public void calcularMontoTotal(){
    .....
}
```

Ejecute el cálculo del total de la factura (el total es la suma de los subtotales) y asigne al mismo al atributo **totalCalculadoFactura**.

- e. Finalmente en la clase **Facturación**, imprima por pantalla la totalidad de los datos de la factura mostrando los datos cargados en la instancia de una forma similar a la siguiente:

<b>Fecha:</b>		XXXXXX			
<b>Numero:</b>		XXXXXX			
<b>Cliente:</b>		XXXXXX			
<b>Código</b>	<b>Nombre</b>	<b>Cantidad</b>	<b>Precio</b>	<b>Descuento</b>	<b>Subtotal</b>
<b>Artículo</b>	<b>Artículo</b>		<b>Unitario</b>		
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX
				<b>Total:</b>	XXXXXX

### 3. Codifique las siguientes clases Java

Clase: Nota

Atributos: catedra (string), notaExamen(double)

Clase: Alumno

Atributos: nombreCompleto(string), legajo (long), ArrayList de objetos Nota

Clase: CargaNotas (clase que tendrá declarado el método main para ejecutar el código.)

Lógica a Implementar:

En la clase CargaNotas debera declarar una variable que permita contener un conjunto de Alumnos:

Ejemplo:

```
ArrayList<Alumno> alumnos = new ArrayList<Alumno>();
```

El algoritmo debe permitir cargar N cantidad de alumnos y para cada alumno N cantidad de Notas. Al finalizar la carga de los alumnos y sus notas mostrar la información cargada y para cada alumno mostrar el promedio de las notas que posee. Valide que se ingrese al menos 1 nota. Agregue en la clase Alumno un método público que calcule el promedio de las notas que posee.

Posible Pseudocódigo básico:

Definir alumnos como Lista

Repetir

Definir alumno como Alumno

Escribir "INGRESE DATOS DEL ALUMNO"

Escribir "INGRESE NOMBRE COMPLETO"

Leer nombreCompleto

Escribir "INGRESE LEGAJO"

Leer legajo

Repetir

Definir nota como Nota

Escribir "INGRESE NOMBRE CATEDRA"

Leer nombreCatedra

Escribir "Nota"

Leer notaExamen

Escribir "DESEA SALIR DE LA CARGA DE NOTAS"

Cargar nota en Lista de Notas

Leer salirNotas

Hasta Que salirNotas

Cargar alumno en Lista alumnos

Escribir "DESEA SALIR DE CARGA DE ALUMNOS"

Leer salirAlumno

Hasta Que salirAlumno

Para alumno en alumnos Hacer

Escribir "Datos Alumno", .....

Para nota en notas Hacer

Escribir "Nota" .....

Fin Para

Escribir "El promedio del alumno es: " promedio

Fin Para

#### 4. Codifique las siguientes clases Java

Clase: Ingrediente

Atributos: nombre (string), cantidad(double), unidad de medida (string)

Clase: Plato

Atributos: nombreCompleto(string), precio (double), esBebida (boolean), ArrayList de objetos Ingrediente

Clase: MenuRestaurant (clase que tendrá declarado el método main para ejecutar el código.)

Lógica a Implementar:

En la clase MenuRestaurant debera declarar una variable "platosMenu" que permita contener un conjunto de Platos que componen el menú de un restaurant.

#### Ejemplo:

```
ArrayList<Plato> platos = new ArrayList<Plato>();
```

El algoritmo debe permitir cargar N platos y para cada plato indicar los N ingredientes que lo componen. Si el plato es de tipo Bebida entonces no se deben solicitar los ingredientes, en cambio si no lo es, será obligatorio que se asigne al menos 1 ingrediente.

Al finalizar la carga de los platos y sus ingredientes mostrar la información cargada, la cual será equivalente al menú del restaurant.

#### Descripción del algoritmo:

Solicito los datos del plato (Ejemplo Pizza Especial, 450 pesos) y sus ingredientes (harina 100 gramos, huevos 2 unidades, queso 300 gramos, jamón 100 gramos, etc) o si es una bebida solo pido el nombre y el precio.

Almaceno el plato anterior en la lista de platos "platosMenu", repito este proceso para tantos platos como desee. Al finalizar muestro la información respetando el siguiente formato ejemplo

-----MENÚ-----

Pizza Especial

Precio: \$ 450

Ingredientes:

Nombre	Cantidad	Unidad de Medida
Queso Muzza	300	gramos

Jamon Cocido        100        gramos

-----  
Vino Tinto Elementos

Precio: \$ 300

-----  
Empanadas Criollas

Precio: \$ 500

Ingredientes:

Nombre	Cantidad	Unidad de Medida
--------	----------	------------------

Picadillo	150	gramos
-----------	-----	--------

Tapa Empanada	1	unidad
---------------	---	--------

-----  
.....repetir.....para el resto de los platos