

LENGUAJE JAVA

Andres Peña Castillo

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA
INGENIERÍA ELECTRÓNICA
PROGRAMACIÓN
2016

TABLA DE CONTENIDO

INTRODUCCIÓN.....	4
OBJETIVOS	6
Objetivo general.....	6
Objetivos específicos	6
CONTENIDO	7
Estructura del software	8
CONCLUSIONES.....	12
REFERENCIAS BIBLIOGRÁFICAS	13

INDICE DE ILUSTRACIONES

<i>Ilustración 1. Estructura de paquetes</i>	8
<i>Ilustración 2. Creación de la clase producto</i>	9
<i>Ilustración 3. Clase archivo</i>	10
<i>Ilustración 4. Clase principal</i>	10
<i>Ilustración 5. Funciones o métodos</i>	11
<i>Ilustración 6. Inventario</i>	11

INTRODUCCIÓN

Actualmente, se presentan todo tipo de problemas relacionados con inventarios; como inventarios físicos que no coinciden con los reportes, inventarios dañados o caducados, o excesos de inventarios, los cuales ocurren en todo tipo de empresas, desde las tiendas de barrio hasta almacenes de clase mundial. (Montaño, 2009). Y muchas veces no se presta la atención necesaria a dichos inconvenientes hasta que representa un problema mayor o una repercusión en las ganancias de la empresa. Ya que, este tipo de problemas pueden acarrear importantes pérdidas monetarias para las organizaciones.

Para tener un proceso de mejora, es importante primero que todo identificar la raíz de estos problemas. Para ello, se han creado diferentes tipos de herramientas informáticas que automatizan la información y hacen más fáciles tareas que antes de estos parecieran eran más complejas. En este orden de ideas, se presentan los software de inventarios, que están diseñados para que el administrador o dueño del negocio, según sea el caso, consulte su estado de existencias, ventas y lo más relevante, sus ganancias, de tal manera que lleve un control minucioso de la contabilidad de su empresa y no se presenten pérdidas ni inconsistencias a la hora que evaluar su rentabilidad.

Por otra parte, es necesario tener en cuenta el tipo de empresa a la cual se va a diseñar el software de inventario ya que no es lo mismo crear un inventario para un supermercado, en el cual, el principal problema podría ser la pérdida del inventario físico o un mal manejo de registro de cuando ingresan o se venden los productos, mientras que para un restaurante, el problema podría ser la caducidad de los insumos, porque se compra más a los proveedores de lo que realmente se vende en el negocio.

Con base en lo anterior, se plantea desarrollar un software de inventario para llevar la contabilidad de un supermercado de barrio, en el cual se han presentado las problemáticas expuestas anteriormente.

Básicamente, en el presente documento se desarrolla la problemática establecida, su solución algorítmica mediante el lenguaje de programación Java y la herramienta de “ID eclipse”, el objetivo principal de la actividad y los objetivos específicos para el diseño del programa, su estructura, el proceso de realización y las conclusiones evidenciadas.

OBJETIVOS

Objetivo general

- Desarrollar un software de inventario que permita mejorar la contabilidad de las existencias, ventas y ganancias de un supermercado.

Objetivos específicos

- Crear la lista de inventarios de los productos existentes dentro del supermercado
- Diseñar la lógica necesaria para la construcción el software
- Diseñar el modulo inventario
- Ejecutar el software en un entorno o sistema operativo

CONTENIDO

El problema planteado en el presente documento, es que, en el supermercado del barrio, a pesar de las altas ventas diarias que reporta, aun no lleva un inventario de contabilidad que especifique, como mínimo costos, ventas y ganancias de cada producto. Por ende, su administrador manifiesta que ha presentado grandes pérdidas monetarias.

Teniendo en cuenta lo anterior, se propone el diseño de un software de inventario, en el cual se especifican los siguientes ítems:

1. **Código del producto:** Se utilizan mínimo 4 caracteres y 4 números para identificar un tipo de producto, por ejemplo: ARRZ0001, ACEI2030.
2. **Existencias:** Son la cantidad de unidades de un mismo producto.
3. **Descripción:** Se refiere a los detalles de los productos, por ejemplo: existe arroz por libras, por kilos, etc.
4. **Costo unitario:** Es la cantidad monetaria que se paga al proveedor para adquirir su producto.
5. **Margen de ganancia:** Es el porcentaje que se desea obtener de ingreso por la venta de un producto (Webaprender, 2016), que, para este caso, va a ser de un 20%
6. **Precio unitario:** Es la cantidad de dinero al que se vende el producto al público en general. Se calcula: $\text{Costo unitario} + (\text{Costo unitario} \times \text{Margen de Ganancia})$
7. **Costo por producto:** Es la cantidad de existencias multiplicada por el costo unitario. Se calcula: $\text{Existencias} \times \text{costo unitario}$.
8. **Ventas por producto:** Es la cantidad de existencias multiplicada por el precio de venta. Se calcula: $\text{Existencias} \times \text{Precio de venta}$.
9. **Ganancia:** Es la cantidad de dinero que se obtiene por vender todas las existencias de un producto. Se calcula: $\text{Ventas por producto} - \text{Costos por productos}$.
10. **Subtotal:** Es la suma de todas las ventas por productos.

11. Iva: Es un impuesto que se cobra por la venta de un producto. Se calcula:

Subtotal x 0.16

12. Total: Es la suma del SUBTOTAL + IVA.

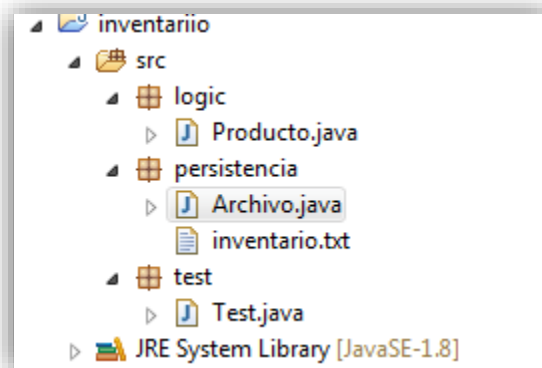
Estructura del software

Para realizar el software de inventario, se trabajó con la herramienta de programación “ID eclipse” y el lenguaje de programación java

A continuación, se presenta la estructura de paquetes y el desarrollo de la lógica del problema planteado:

1. Primero que todo, se realiza la estructura de paquetes que va a ser utilizada para el desarrollo del software de inventario

Ilustración 1. Estructura de paquetes



Fuente: Autor.

2. En segundo lugar, se realiza la creación de la clase producto, la cual va a almacenar los productos que tiene el supermercado

Ilustración 2. Creación de la clase producto

```
public class Producto {  
    private String codigo;  
    private int existencias;  
    private String descripcion;  
    private double costoUnitario;  
    private String margenGanacia;  
    private double precioUnitario;  
    private double costoProducto; //adentro  
    private double ventasProducto; //adentro  
    private double ganancia; //resta  
  
    public Producto(String codigo, int existencias, String descripcion,  
                    double costoUnitario, String margenGanacia, double precioUnitario,  
                    double costoProducto, double ventasProducto, double ganancia) {  
        super();  
        this.codigo = codigo;  
        this.existencias = existencias;  
        this.descripcion = descripcion;  
        this.costoUnitario = costoUnitario;  
        this.margenGanacia = margenGanacia;  
        this.precioUnitario = precioUnitario;  
        this.costoProducto = costoProducto;  
        this.ventasProducto = ventasProducto;  
        this.ganancia = ganancia;  
    }  
    public String getCodigo() {  
        return codigo;  
    }  
    public void setCodigo(String codigo) {  
        this.codigo = codigo;  
    }  
    public int getExistencias() {  
        return existencias;  
    }  
    public void setExistencias(int existencias) {  
        this.existencias = existencias;  
    }  
    public String getDescripcion() {  
        return descripcion;  
    }  
}
```

Fuente: Autor.

3. Luego se realiza la clase que lee todos los productos de un archivo local, como se muestra en la siguiente ilustración:

Ilustración 3. Clase archivo

```
import java.io.BufferedReader;

public class Archivo {
    private FileReader file;
    private BufferedReader buffer;
    public Archivo(String path) {
        // TODO Auto-generated constructor stub
        try {
            file = new FileReader(path);
        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        buffer = new BufferedReader(file);
    }

    public ArrayList<Producto> leerTodo() throws NumberFormatException, IOException {
        ArrayList<Producto> productos = new ArrayList<Producto>();
        String cadena;
        Producto producto = null;
        while ((cadena = buffer.readLine()) != null) {
            String [] arrayLine = cadena.split(",");
            producto = new Producto(arrayLine[0], Integer.parseInt(arrayLine[1]), arrayLine[2], Double.parseDouble(arrayLine[3]), arrayLine[4], Double.parseDouble(arrayLine[5]));
            productos.add(producto);
        }
        buffer.close();
        return productos;
    }
}
```

Fuente: Autor.

4. A continuación, se realiza la clase principal para ejecutar la lógica del inventario realizado:

Ilustración 4. Clase principal

```
public static void main(String[] args) {
    Archivo archivo = new Archivo("src/persistencia/inventario.txt");
    try {
        ArrayList<Producto> listaProductos = archivo.leerTodo();
        System.out.println("CODIGO\tEXISTENCIAS\tCOSTO UNITARIO\tMARGEN DE GANANCIA\tPRECIO UNITARIO\tCOSTOS X PRODUCTO");
        for (int i = 0; i < listaProductos.size(); i++) {
            Producto producto = listaProductos.get(i);
            System.out.println(producto.getCodigo()+"\t"+
                producto.getExistencias()+"\t"+producto.getCostoUnitario()+"\t"+
                producto.getMargenGanancia()+"\t"+producto.getPrecioUnitario()+"\t"+
                producto.getCostoProducto()+"\t"+producto.getVentasProducto());
        }
        double subtotal = calcularSubtotal(listaProductos);
        double iva = iva(subtotal);
        System.out.println();
        System.out.println("SUBTOTAL: "+subtotal);
        System.out.println("IVA: "+iva);
        System.out.println("TOTAL: "+total(subtotal, iva));
    } catch (NumberFormatException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    while(true);
}
```

Fuente: Autor.

5. También se deben establecer las funciones o métodos que permitan calcular el subtotal, IVA y total, los cuales retornan un tipo de dato double

Ilustración 5. Funciones o métodos

```

public static double calcularSubtotal(ArrayList<Producto>lista){
    double aux=0;
    for (int i = 0; i < lista.size(); i++) {
        Producto producto=lista.get(i);
        aux+=producto.getVentasProducto();
    }
    return aux;
}

public static double iva(double subtotal){
    return 0.16*subtotal;
}

public static double total(double subtotal,double iva){
    return subtotal+iva;
}
  
```

Fuente: Autor.

6. Finalmente, se genera el inventario del supermercado

Ilustración 6. Inventario

CODIGO	EXISTENCIAS	COSTO UNITARIO	MARGEN DE GANANCIA	PRECIO UNITARIO	COSTOS X PRODUCTO	VENTAS X PRODUCTOS	GANANCIA	DESCRIPCION
SABR0800	13	27.0	20%	32.4	351.0	421.2	70.2	SABRITAS 35gr
REFR0804	19	14.0	20%	16.8	266.0	319.2	53.2	REFresco LATA 35ML
ATUN7878	17	13.0	20%	15.6	221.0	265.2	44.2	ATUN 300GR
ACEI4715	14	12.0	20%	14.4	168.0	201.6	33.6	ACEITE 1-2-3 1 LITRO
GALL9898	17	9.0	20%	10.8	153.0	183.6	30.6	GALLETAS 50GR
TORT2016	15	13.0	20%	15.6	195.0	234.0	39.6	TORTILLAS 1KG
PLAT1478	11	13.0	20%	9.6	88.0	105.6	17.6	TORTILLAS 1KG
FRIJ4545	12	15.0	20%	18.0	180.0	216.0	36.0	LATA DE FRIJOL
ARRO9696	10	12.0	20%	14.4	120.0	144.0	24.0	ARROZ 1KG EL VALLE
AGUA2020	14	7.0	20%	8.4	98.0	117.6	19.6	AGUA PURIFICADA CRISTAL
SUBTOTAL: 2207.9999999999999								
IVA: 353.27999999999999								
TOTAL: 2561.2799999999999								

Fuente: Autor.

CONCLUSIONES

- Con el desarrollo de este trabajo, se refinaron más los conceptos de lenguaje de programación Java para la realización y el desarrollo del software inventario.
- Un software es una buena herramienta para dar solución a un problema como el mal manejo de los inventarios.
- También se puede concluir que el lenguaje de programación Java es portable, lo cual se refiere a que se puede ejecutar en cualquier sistema operativo ya sea Windows, Mac, Linux, entre otros, con tan solo instalar Java.
- El IDE eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto, es decir que cualquier persona lo puede utilizar sin que le genere un costo, además de ser uno de los mejores IDE para Java.

REFERENCIAS BIBLIOGRÁFICAS

Montaño, R. (2009). Diseño de un sistema de control del inventario para la empresa inversiones Miwill, C.A., Maturin Estado Monagas, 3. Retrieved from <http://www.monografias.com/trabajos75/sistema-control-inventario-empresa-inversiones/sistema-control-inventario-empresa-inversiones2.shtml>

Webaprender. (2016). Proyecto Excel. Inventario para una tienda de abarrotes. Recuperado de: <http://www.webaprender.com/informatica/excel/proyecto-excel-inventario-una-tienda-abarrotes/>