

Trabajo Principios de lógica y algoritmos
de programación
Parte 2.

Nombre: Miguel Ángel González Jaimén.

Carrera: Ingeniería en Computación e Informática.

Profesor: Simón Muñoz.

Sección: 8

Introducción

En este informe se presentará en detalle el desarrollo de una aplicación básica que permite llevar el control de un stock predeterminado de tres productos. Bajo funciones sencillas lograre crear una venta paso a paso, para luego generar la facturación en función de las elecciones y descontar las cantidades ingresadas del stock.

El proyecto será realizado con las funciones y métodos explicados en clases, en base a esto, el desarrollo de la aplicación es un desarrollo introductorio a Java, donde primordialmente se busca que funcione según lo solicitado, con un uso del código simple y practico.

Lo solicitado.

Considerar que se requiere un sistema que sea capaz de manejar las ventas de diversos productos. Más específicamente los productos son:

Producto	Código	Cantidad inicial en bodega (cajas)	Precio por caja en Ch\$
Peras	AX10	10.000	10.000
Manzanas	AX20	1.000	5.000
Uvas	AX30	20.000	40.000

Considerar que por cada compra se tiene un descuento por volumen, dada la siguiente tabla:

Consideraciones:

$$\left\{ \begin{array}{ll} \textit{Venta} < 1000 \textit{ cajas} & , \quad 0\% \textit{ descuento} \\ 1.000 \leq \textit{Venta} < 5.000 & , \quad 5\% \textit{ descuento} \\ \textit{Venta} \geq 5.000 & , \quad 10\% \textit{ descuento} \end{array} \right.$$

1. El sistema pregunta si se desea vender o ver el stock en bodega.
2. Si seleccionar vender, entonces se pide la cantidad de cajas a comprar para los tres productos.
3. Se debe mostrar: el total sin descuento por cada producto, el total de descuento aplicado a cada producto y el total a pagar.
4. Si selecciona stock en bodega, entonces para los tres productos mostrará la cantidad en stock.
5. Se debe mostrar un mensaje que indique el caso en que la venta supera a stock en bodega.
6. No debe desarrollar ningún tipo de validación a excepción de la indicada en el punto 5.

Desarrollo.

El desarrollo del proyecto lo realice desglosando lo solicitado en 3 partes; venta, facturación y stock. Creando un método por cada parte, los cuales son llamados en el método main según la opción indicada, de la misma forma este informe será realizado método por método buscando una comprensión precisa del código aplicado.

Inicio:

Para comenzar inicializo las variables que serán ocupadas en más de un método, directamente en la clase;

```
public class Solemne {  
  
    public static int sp;//sock  
    public static int sm;  
    public static int su;  
  
    final public static int PP = 10000;//precios $cl  
    final public static int PM = 5000;  
    final public static int PU = 40000;  
  
    public static int cp = -1;// cantidad negativa para validar  
    public static int cm = -1;  
    public static int cu = -1;  
}
```

MAIN:

Declaración de variables;

En main declare el stock inicial del cual se generara el descuento correspondiente a las ventas y a su vez declaro una variable i en "0" que utilizare como validador del proceso general.

```
public static void main(String[] args) throws IOException {  
  
    java.io.BufferedReader    br        =        new        java.io.BufferedReader(new  
    InputStreamReader(System.in));  
  
    sp = 10000; //cantidad inicial  
    sm = 1000;  
    su = 20000;  
  
    int i = 0; //validador
```

Proceso general;

Para el proceso general utilicé la función while, la cual me permite mantener el programa activo mientras no se escoja la opción salir (3).

```
while (i != 3) { Proceso }
```

Menú;

Para la estética del menú realice varias impresiones en pantalla y luego solicite la opción, con las librerías `IOExceptions` e `InputStreamReader` importadas dentro de `package`, para validar la opción escogida recurrí a la función `switch` la cual me permitió realizar la elección de alternativas de una forma ordenada y práctica.

```

System.out.println("-----");
    System.out.println("escoja una opcion");
    System.out.println("-----");
    System.out.println("1-----Venta");
    System.out.println("2-----Stock");
    System.out.println("3-----Salir");
    System.out.print("Opcion----->");
    int x = Integer.parseInt(br.readLine());
    System.out.println("");
    System.out.println("");

```

```

switch (x) {
    case 1:
        venta(); //metodo venta
        facturacion(); //metodo facturación

        break;
    case 2:
        stock();//metodo stock

        break;
    case 3:
        i = 3;// salir
        break;
    default:
        System.out.println("opción no valida");
        break;
    //FIN MENU
}

```

Luego de switch inicio una variable llamada xx en 0 la cual utilizo para ordenar la visualización de la aplicación y así al terminar un proceso me pedirá ingresar 1 para continuar.

```

Switc(x){ menu}
int xx = 0;//validacion para continuar
while (xx != 1) {
    System.out.println("1 para continuar");
    xx = Integer.parseInt(br.readLine());
}

```

Venta:

El método venta lo ordene con tres while (uno para cada producto), dentro de los cuales se indica el producto a ingresar y se solicita una cantidad, en función de la

cantidad ingresada realiza las opciones disponibles, en el caso en que la petición “**cp**” sea mayor que la cantidad de productos en stock “**sp**” imprimirá una alerta indicando la situación actual del stock y pedirá nuevamente una cantidad en el caso contrario descontara la cantidad solicitada “**cp**” y continuara con el proceso, luego de tener valores posibles en los tres productos, dará paso al método facturación.

En este caso mostrare el proceso realizado para uno de los productos ya que el proceso es idéntico para los tres productos (cambiando las variables por cada producto).

```
//PERAS
while (cp < 0) {
    System.out.println("Cajas de peras");
    cp = Integer.parseInt(br.readLine()); //cantidad de peras
    if (cp > sp) {
        System.out.println("la venta supera el stock disponible " + sp);
        cp = -1;
    } else {
        sp = (sp - cp);
    }
}
```

Facturación:

El método facturación seguramente es el método más complejo de todos ya que aquí es donde más cálculos debe realizar la aplicación. Para comenzar calculamos todos los valores brutos de la venta considerando, en el caso de las peras que “**bp=(cp*PP)**”, donde “bp” es el valor bruto de las peras, “cp” es la cantidad recogida en la venta y PP es el precio declarado directamente en la clase contando con este cálculo iniciamos las variables “tp, tm, tu” dentro de las cuales calcularemos los totales parciales, vale decir, “total bruto menos el descuento correspondiente”. Teniendo esto comenzamos la facturación por producto donde analizamos la cantidad solicitada para aplicar el descuento correspondiente según lo solicitado, en un principio el resultado del total bruto menos el descuento se guardara en una variable doblé temp la cual será convertida a entero (int) y guardada como tp, tm o tu según corresponda.

El último paso de la facturación será mostrar los resultados de una forma intuitiva y calcular el total a pagar en base a la suma de los totales parciales.

```
public static void facturacion() {
    int bp =(cp*PP); //totales brutos
    int bm =(cm*PM);
    int bu =(cu*PU);
    int tp; //totales parciales
    int tm;
    int tu;
    System.out.println("");
    System.out.println("");
    System.out.println("Facturación");
    System.out.println("-----");
    //peras
    System.out.println("Peras");
```

```

    if (cp<1000){
        System.out.println("sin descuento");
        tp=bp;
    }
    else if(cp>=1000 && 5000>cp){
        System.out.println("descuento 5%");
        double temp=bp*0.95;
        tp=(int)temp;
    }
    else{
        System.out.println("descuento 10%");
        double temp=bp*0.90;
        tp=(int)temp;
    }
    System.out.println("Total bruto .....$"+bp);

//manzanas
    System.out.println("");
    System.out.println("Manzanas:");
    if (cm<1000){
        System.out.println("sin descuento");
        tm=bm;
    }
    else if(cm>=1000 && 5000>cm){
        System.out.println("descuento 5%");
        double temp=bm*0.95;
        tm=(int)temp;
    }
    else{
        System.out.println("desuento 10%");
        double temp=bm*0.90;
        tm=(int)temp;
    }
    System.out.println("Total bruto .....$"+bm);

//manzanas
    System.out.println("");
    System.out.println("Uvas");
    if (cu<100){
        System.out.println("sin descuento");
        tu=bu;
    }
    else if(cu>=1000 && 5000>cu){
        System.out.println("descuento 5%");
        double temp=bu*0.95;
        tu=(int)temp;
    }
    else{
        System.out.println("descuento 10%");
        double temp=bu*0.90;
        tu=(int)temp;
    }

```



```

    }
    System.out.println("Total bruto .....$" + bu);

    //datos
    int t = (tp + tm + tu); //total a pagar
    System.out.println("");
    System.out.println("Total a pagar.....$" + t + ".-");
    System.out.println("");
    System.out.println("-----");
    System.out.println("");
    System.out.println("");
    cp = -1;
    cm = -1;
    cu = -1;
}

```

Stock:

Así como la facturación es un método complejo el método stock debe ser el más sencillo ya que se basa completamente en mostrar variables con un orden lógico y los más intuitivamente posible, para esto recurrí a las variables obtenidas en la venta al descontar los productos del stock.

```

//STOCK
public static void stock() throws IOException {
    java.io.BufferedReader br = new java.io.BufferedReader(new
    InputStreamReader(System.in));

    final String OP = "AX10"; //codigos
    final String OM = "AX20";
    final String OU = "AX30";

    System.out.println("");
    System.out.println("");
    System.out.println("Stock");
    System.out.println("-----");
    System.out.println("- Prudcto    Codigo Precio  Cantidad");
    System.out.println("- peras....." + OP + "....$" + PP + "....." + sp + ".-");
    System.out.println("- Manzanas....." + OM + "....$" + PM + "....." + sm + ".-");
    System.out.println("- Uvas....." + OU + "....$" + PU + "....." + su + ".-");
    System.out.println("-----");
    System.out.println("");
}

```

Conclusión

Al leer por primera vez lo solicitado se veía bastante complejo, pero gracias a los conceptos y ejercicios realizados en clases, basto con comenzar a escribir para que el código comenzara a fluir fácilmente y todo comenzara a funcionar, claramente existieron detalles a investigar, pero en java se pueden descubrir comunidades y documentaciones fácilmente, lo que hace que la autonomía del aprendizaje sea bastante dinámica, y esto para alguien que está en esta carrera se vuelve muy entretenido, en mi caso el código inicial tiene bastantes diferencias con lo que entrego en este informe y es que cada vuelta que le daba era un cambio para simplificar el código y seguramente al leerlo nuevamente seguiré encontrando detalles que mejorar, pero para eso me quedan varios semestres más de aprendizaje.

Miguel A. González J.

Diagrama de flujo - General

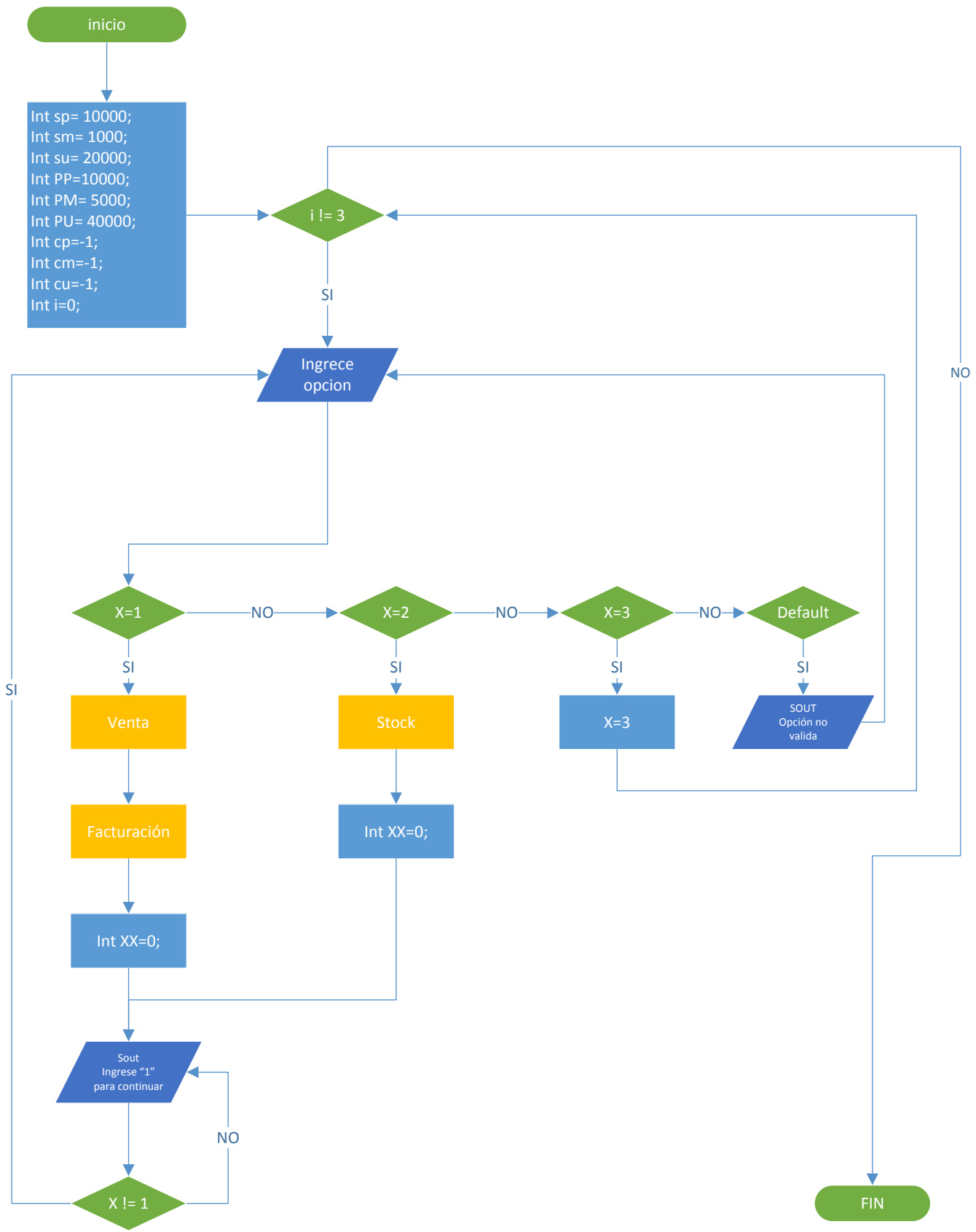


Diagrama de flujo - Caso 1 - VENTA

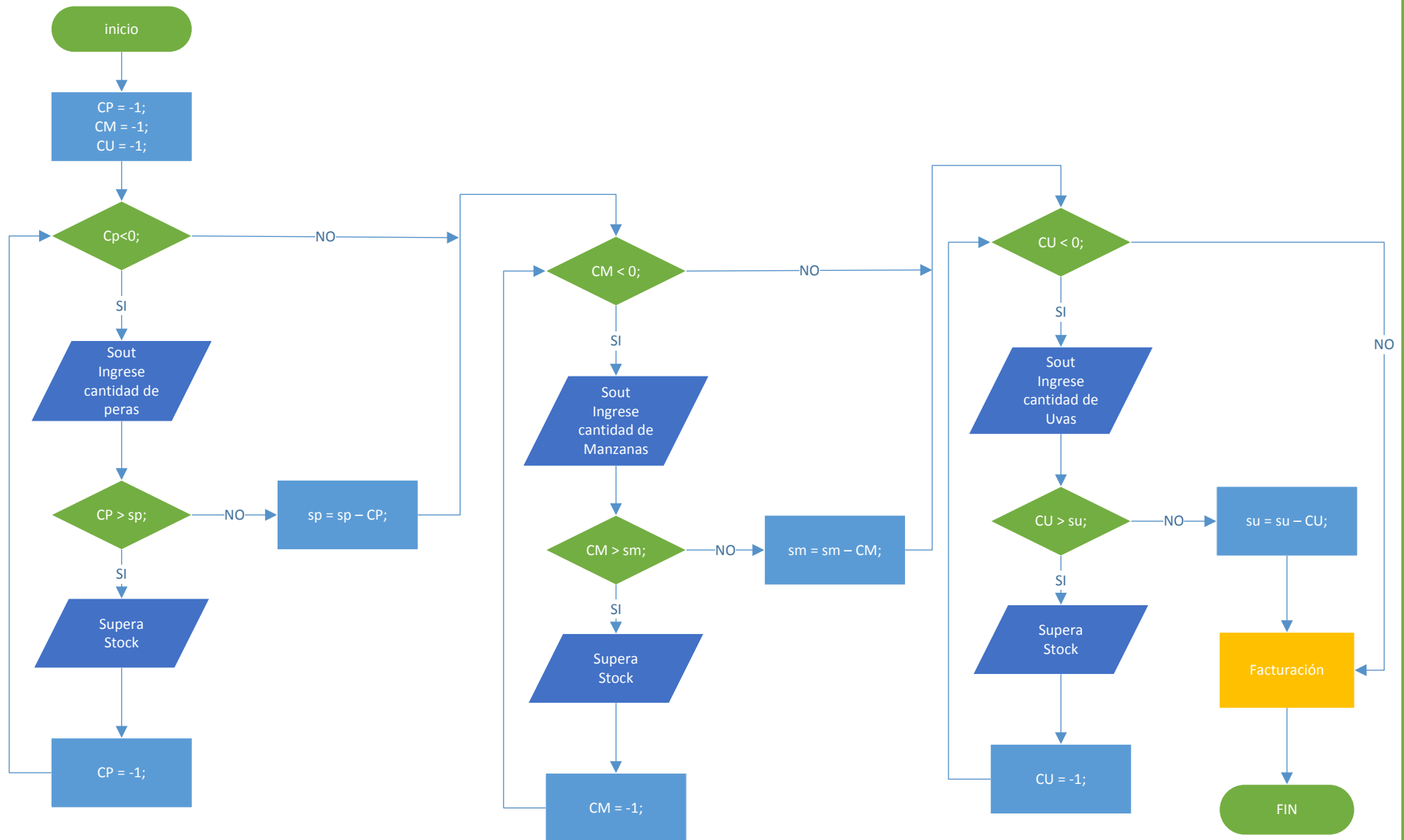


Diagrama de flujo - Caso 1 - FACTURACIÓN

