

*PROGRAMACIÓN I (PUI)*  
**TRABJO PRÁCTICO 2: MODULARIDAD**

**Ejercicio 1.**

**Diseñar el árbol de jerarquía de módulos aplicando los principios de Programación Modular. Expresar en Pseudocódigo las estructuras de datos a utilizar**

**1)** El supermercado “*Día*” desea que se le confeccione un programa que le brinde las siguientes opciones:

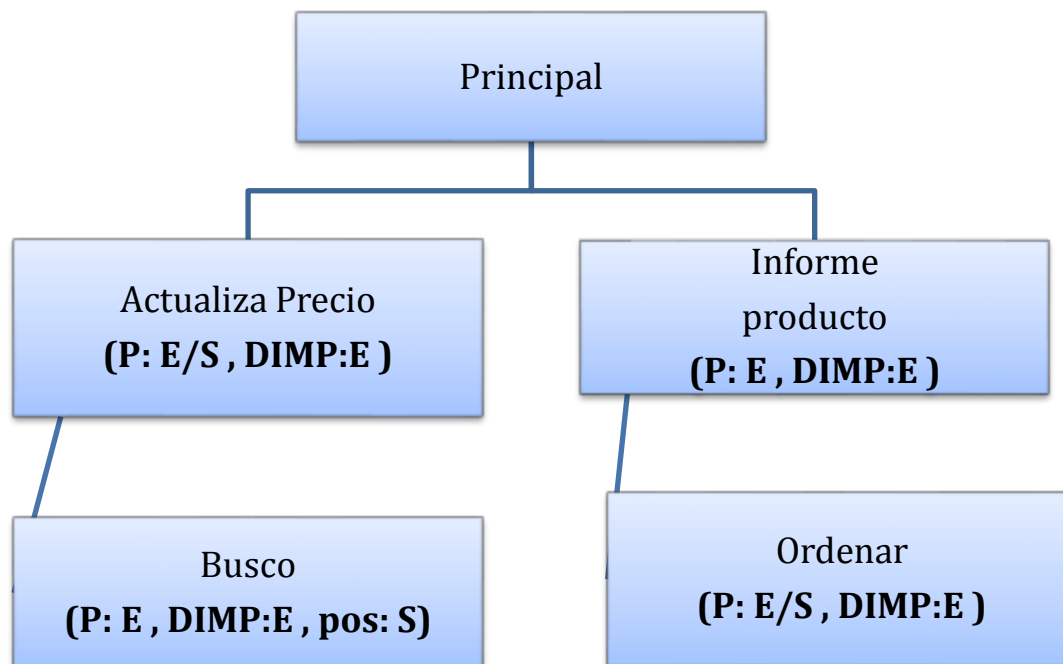
- 1.- Actualización de precios
- 2.- Informe de Productos
- 3.- Finalizar el programa

Para ello dispone de la siguiente información:

- ✓ El arreglo **P[DIM]**, de tipo registro contiene los siguientes datos: Código Producto, Descripción, Cantidad existente y Precio.
- ✓ En el módulo *Actualización de precios*: se lee un código de producto y el porcentaje a incrementar, si el producto existe deberá actualizar su precio, en caso contrario mostrará el mensaje “*Producto Inexistente*”
- ✓ En el módulo *Informe de productos*: realizar un listado con el siguiente formato de aquellos productos cuyo monto en stock sea mayor que \$100.000

**Informe de productos**

Código	Descripción	Monto en stock
101	HARINA	550.000
123	FIDEOS	150.000
154	ACEITE	320.000



Pseudocódigo de la solución

**PROGRAMA Supermercado DIA**  
**Pseudocódigo del Registro**

RegP registro

CP: entero  
Desc: Cadena  
Cex:cadena  
Prec:real

**VARIABLES:** Fin de registro  
Opción: entero  
P[] RegP  
DIM:entero

**REPEAT**

MOSTRAR "1.- Actualización de Precios"

MOSTRAR "2.- Informe de Productos"

MOSTRAR "3.- Finalizar el Programa"

LEER Opcion

**CASE** (Opcion)

Opcion = 1: **ActualizaPrecio(P, DIM)**

Opcion = 2: **InformeProductos (P, DIM)**

**ENDCASE**

**UNTIL** Opcion = 3

**FIN**

**PROCESO ActualizaPrecio (P:E/S , DIMP:E )**

// Actualiza el precio de un producto ingresado, si se encuentra

Escribir "Ingrese el producto a buscar"

Leer prod

I=Busco (prod,P,DIMP)

Si I=-1 entonces

Escribir "No se encontró"

Sino

Escribir "Ingrese el porcentaje a incrementar"

Leer Inc

$P[i].Prec = P[i].Prec * (1 + inc/100)$

Fin Si

**FIN**

**PROCESO InformeProductos (P[:E , DIMP:E )**

// Muestra un listado de los productos que cumplen con la condición especificada

**Ordenar(P[],DIMP)**

Escribir "Informe de productos"

Escribir "Código"	Descripción	Monto en Stock"
-------------------	-------------	-----------------

Para i=0 hasta DIMP-1

Monto = P[i].cex\* P[i].prec

Escribir P[i].CP

Escribir P[i].Desc

Escribir Monto

**FIN PARA**

**FIN**

**PROCESO Ordenar (P[:E/S , DIMP:E )**

// ordena el arreglo P en forma ascendente por código de producto)

AUX : RegP

//Selección ascendente

I = 0

**MIENTRAS** I < DIMP -1

Pos = I

```

        J = I + 1
        MIENTRAS J <= DIMP-1
            SI P[J].CP < P[Pos].CP
                ENTONCES
                    Pos = J
            FIN SI
            J = J + 1
        FIN MIENTRAS
        AUX = P[Pos]
        P[Pos] = P[I]
        P[I] = AUX
        I = I + 1
    FIN MIENTRAS
FIN

```

**Función Busco (CodigoProd:E , P[]:E , DIMP:E ):Entero**

```

// Función que busca el código de producto en el arreglo P[]
//y retorna la posición encontrada como un entero positivo ó -1 si no existe)
//Las instrucciones: Escribir "Ingrese el producto a buscar" y "Leer prod", no se realizan en este módulo sino que se
//ejecutan en el llamador y luego se recibe el producto en CodigoProd como parámetro de entrada para este módulo
Resultado=-1
I=0
Mientras i<DIMP and Resultado=-1
    Si P[i].CP=CodigoProd entonces
        Resultado = i
    Fin Si
    I=i+1
Fin Mientras
Busco=Resultado
FIN

```