



Departamento de Lenguajes
y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

LABORATORIO DE PROGRAMACIÓN II

E.T.S.I.Informática. 1º. 06-09-01

Duración del Examen: 4 horas

APELLIDOS _____ NOMBRE _____

GRUPO(A/B/C/D) _____ ESPECIALIDAD(Gestión/Sistemas) _____ ORDENADOR _____

El nuevo supermercado HiperMod nos ha solicitado la elaboración del software para la gestión y manejo de sus cajas. Para ello deberemos implementar y usar el módulo MBolsa cuya definición se adjunta, y realizar un programa en Modula-2 llamado **HIPER.MOD** con las siguientes opciones de menú:

```
Programa Realizado por: Apellido1 Apellido2, Nombre
                        Especialidad: Gestión/Sistemas
                        Grupo: A/B/C/D
                        Máquina: XXX
```

```
HiperMod
=====
```

- A. Introducir Producto en la Bolsa.
- B. Mostrar el contenido de la Bolsa.
- C. Sacar un Producto de la Bolsa.
- D. Imprimir Ticket.
- E. Compra por Internet.
- X. Salir del programa

Descripción de Opciones:

- A. Introducir un Producto en la Bolsa: Solicita al usuario la referencia y nombre del producto, introduciéndolo en la bolsa. En caso de que la bolsa estuviera llena se informará del error y no se insertará.
- B. Mostrar el contenido de la Bolsa: Muestra por pantalla el contenido de la bolsa (referencia y nombre), sin modificar el contenido de la misma.
- C. Sacar un Producto de la Bolsa: Solicita al usuario la referencia de un producto y lo elimina de la bolsa.
- D. Imprimir Ticket: Muestra por pantalla el contenido de la bolsa (referencia, nombre y precio), a medida que va vaciando la misma. Al final de todos los productos se imprime el total de la compra. Los precios de los productos se obtendrán del fichero "PVP.TXT". El fichero PVP.TXT es un fichero que contiene un registro por línea con el formato: <Referencia> <Precio>

Ejemplo: PVP.TXT

```
125 98
424 2995
321 6708
010 78778
.....
```

- E. Compra por Internet: Se solicitará el nombre de un fichero en el que se encuentran los productos solicitados por el cliente en el formato siguiente: <nombre del producto>#<Referencia>

Ejemplo: Cliente.txt
Patatas Fritas#125
CD Kill'em All#424
.....

Se leerá dicho fichero generándose una bolsa con el contenido del mismo, de la cual se generará un ticket en el fichero de texto TICKET.OUT (si dicho fichero existe se borra su contenido) con el mismo formato que la impresión de tickets por pantalla.

- X. Salir del programa: Se pedirá confirmación antes de salir del programa y en caso afirmativo se liberarán todos los recursos que hayan sido reservados.

Módulo de definición de una bolsa QUE NO PODRÁ SER MODIFICADO NI SE LE PODRÁ AÑADIR O QUITAR FUNCIONALIDAD.

```
DEFINITION MODULE MBolsa;
```

```
CONST
```

```
    MAXCAD = 20;
```

```
    FINCAD = CHR(0);
```

```
TYPE
```

```
    TBolsa;
```

```
    TCadena = ARRAY [1..MAXCAD] OF CHAR;
```

```
    TElemento = RECORD
```

```
        referencia: CARDINAL;
```

```
        producto: TCadena;
```

```
    END;
```

```
PROCEDURE CrearBolsa():TBolsa;
```

```
(* Crea una Bolsa Vacía. *)
```

```
PROCEDURE BolsaVacía(b:TBolsa):BOOLEAN;
```

```
(* Comprueba si la bolsa que se le pasa como parámetro está vacía. *)
```

```
PROCEDURE BolsaLlena(b:TBolsa):BOOLEAN;
```

```
(* Comprueba si la bolsa que se le pasa como parámetro está llena. *)
```

```
PROCEDURE MeterBolsa(VAR b:TBolsa; e:TElemento);
```

```
(* Mete el Elemento 'e' si la bolsa no está llena. En caso de estar llena  
no realiza nada. Es responsabilidad del usuario de este módulo el control  
de que la bolsa esté o no llena.  
*)
```

```
PROCEDURE SacarBolsa(VAR b:TBolsa; VAR e:TElemento);
```

```
(* Saca un Elemento 'e' CUALQUIERA de la bolsa si la bolsa no está vacía.  
En caso de estar vacía no realiza nada y 'e' contendrá basura.  
Es responsabilidad del usuario de este módulo el control  
de que la bolsa esté o no vacía.  
*)
```

```
PROCEDURE DestruirBolsa(VAR b:TBolsa);
```

```
(* Destruye la bolsa que se le pasa como parámetro *)
```

```
END MBolsa.
```

Se valorará:

1. La corrección del programa.
2. Una buena modularización (uso de módulos/procedimientos/funciones)
3. Uso de tipos adecuados.
4. La ausencia de efectos laterales.
5. La legibilidad del código.
6. La presencia y claridad de mensajes de salida resultado de cada operación.

Notas:

1. Se puede utilizar cualquier subrutina de los módulos estándares de TOPSPEED **EXCEPTO LA LIBRERÍA STR**
2. Es obligatorio trabajar en el directorio **C:\HIPER**. Si no existe se creará. El programa principal se llamará **HIPER.MOD**
3. **Para Aprobar** deberá ser correcta la definición de tipos, la modularización y funcionar **CORRECTAMENTE** las **Opciones A,B, C, D y X** del menú.
4. La opción E del menú servirá para obtener más de aprobado.