

MEMORIA: PRÁCTICA GRUPAL

En la realización de cada uno de los subprogramas, han intervenido cada uno de los componentes de este grupo mediante una puesta en común.

Guillermo Navas Gracia
Iván Pérez Huete
Daniel Peña Martínez
Jesús Prieto García

`-PROCEDURE mostrarMenu;`

`-PROCEDURE altaComponente (VAR datos:tAlmacenComponentes);`

Este subprograma permite al usuario la posibilidad de dar de alta un componente, para ello, si no está completo el almacén, pedirá todos los datos acerca del componente.

`-PROCEDURE Eliminar (VAR datos:tAlmacenComponentes ;
idem:string);`

Este subprograma busca en el almacén de componentes, un componente específico. En este caso lo hemos hecho para que lo busque por el nombre del identificador del componente en cuestión.

`-PROCEDURE configurarOrdenador(VAR
datos:tAlmacenComponentes ; VAR datospcc:tAlmacenPcs);`

Este subprograma permite dar de alta (configurar) un ordenador. Esta acción se realiza de la siguiente forma. Primero se deben introducir los componentes y características requeridas. A continuación se calcula el precio total de este ordenador teniendo en cuenta sus componentes. Por último, si es posible, se añadirá el ordenador al almacén.

`-PROCEDURE mostrarVentaComponente (numero:integer; VAR
datos:tAlmacenComponentes);`

Subprograma que muestra la suma de dinero recaudada, es decir, las ventas totales.

`-PROCEDURE venderComponente (iden:tIdentificador; VAR
datos:tAlmacenComponentes; VAR datos2:tTienda);`

Este subprograma permite vender un componente mediante su identificador. Si la venta se realiza con éxito, el número de ventas se incrementa y dicho artículo se retira del almacén.

```
-PROCEDURE modificarComponente (idMod:string; VAR  
datos:tAlmacenComponentes);
```

Este subprograma ofrece la posibilidad de editar la información de un componente específico. Para acceder a dicho objeto se introduce el identificador correspondiente y el campo que se desea modificar. El almacén de componentes guardará los datos actualizados de nuevo.

```
-PROCEDURE venderOrdenador (id:tIdentificador; VAR  
datosOrd:tAlmacenPcs; tienda:tTienda);
```

Este subprograma es similar a venderComponente.

```
-PROCEDURE mostrarOrdenadores (dato:tAlmacenPcs);
```

Subprograma que muestra todos los ordenadores disponibles del almacén de más barato a más caro.

```
-PROCEDURE mostrarComponentes (datos:tAlmacenComponentes);
```

Subprograma que muestra todos los componentes disponibles en el almacén ordenadamente.

```
-PROCEDURE guardarDatosBin (VAR fich:tFicheroComponentes;  
datos:tAlmacenComponentes);
```

Este subprograma se divide en dos. Por un lado se escribe dentro de un fichero binario denominado 'componentes.dat', todos los componentes del almacén. Por otro lado, ocurre lo mismo con los ordenadores almacenados, en un archivo denominado 'ordenadores.dat'.

```
-PROCEDURE guardarDatosText (datosComp:tAlmacenComponentes;  
datosPcs:tAlmacenPcs; VAR fichComp:text ; VAR  
fichPcs:text);
```

Al igual que en el anterior subprograma, se escriben todos los datos de los componentes y los ordenadores, en este caso en un archivo de texto.

```
-PROCEDURE cargarDatosBin (VAR  
datosComp:tAlmacenComponentes; VAR datosPcs:tAlmacenPcs;  
VAR fichComp:tFicheroComponentes; VAR fichPcs:tFicheroPcs);
```

Carga todos los datos respectivos a componentes y ordenadores desde un archivo binario. Estos nuevos datos reemplazarán a los que ya estaban presentes en la memoria principal del programa.

```
-PROCEDURE cargarDatosText ;
```

Al igual que en el anterior apartado, se cargan los datos pero esta vez desde un archivo de texto.