

PRÁCTICA 1 RPC

Enunciado

La práctica consistirá en el desarrollo e implementación de una aplicación cliente-servidor para la gestión de almacenes de productos. Cada cliente podrá gestionar un solo almacén al mismo tiempo, aunque el mismo almacén puede estar gestionado por más de un cliente.

El almacenamiento y la gestión de la información relativa a los productos del almacén se realizará mediante ficheros y memoria dinámica. Para ello se diseñará el siguiente menú principal:

```
----- Menú Almacenes ----- Nombre Almacén

1.- Crear un almacén vacío.
2.- Abrir un fichero de almacén.
3.- Cerrar un almacén.
4.- Guardar Datos.
5.- Listar productos del almacén.
6.- Añadir un producto.
7.- Actualizar un producto.
8.- Consultar un producto.
9.- Eliminar un producto.
0.- Salir.
```

En el título del menú Almacenes, debe aparecer el nombre del almacén que se esté gestionando en ese momento o nada si no se está gestionando ningún almacén.

En el servidor existirá un conjunto de ficheros con los datos de cada almacén. Estos ficheros podrán ser creados o bien cargados a memoria en vectores dinámicos. Los servidores siempre trabajarán con los datos almacenados en la memoria y puntualmente dichos datos podrán ser volcados nuevamente a ficheros.

En todo momento, las distintas opciones del menú para la gestión de los almacenes deben controlar si el almacén está creado y/o cargado (opciones de la 3 a la 9), si los productos solicitados están o no en el almacén, etc. Además, se debe mostrar al final de cada opción del menú el estado de su ejecución, es decir, un mensaje de error que especifique qué ha impedido la realización de la opción o bien un mensaje que indique que ha finalizado satisfactoriamente.

Todas estas gestiones se realizarán en el servidor mediante la invocación de procedimientos remotos (RPC) por los clientes.

Las distintas opciones del menú tienen la siguiente funcionalidad:

- 1.- Crear un almacén vacío => Solicita los datos del almacén (nombre, dirección y nombre de fichero). Verificará que el fichero almacén no está creado previamente y en dicho caso creará un fichero físico vacío (sin productos) además de la memoria dinámica correspondiente para poder trabajar con dicho almacén. Si el almacén ya está creado por otro cliente podrá compartirlo y utilizarlo.

- 2.- Abrir un fichero almacén => Solicita un fichero de almacén previamente creado y lo carga en memoria para su utilización. Si previamente ya estaba cargado, el cliente podrá compartirlo y utilizarlo.
- 3.- Cerrar un almacén => Vuelca toda la información de la memoria dinámica al fichero. Si un almacén es compartido por varios clientes, el último cliente después de volcar la información al fichero eliminará la correspondiente memoria dinámica.
- 4.- Guardar Datos => Volcará todos los datos del almacén almacenados en memoria al fichero.
- 5.- Listar productos del almacén => Muestra en pantalla todos los datos de cada producto del almacén de manera tabular.
- 6.- Añadir un producto => Solicita los datos de un producto y lo añadirá al almacén si este producto no existe.
- 7.- Actualizar un producto => Debe solicitar el producto a actualizar. Una vez localizado, por cada dato del producto debe mostrarlo y preguntar si se desea modificarlo, si se responde afirmativamente se solicita un nuevo valor. Una vez terminado la entrada de datos se actualiza el producto del almacén con los nuevos datos.
- 8.- Consultar un producto => Debe solicitar el producto a mostrar. Una vez localizado debe mostrar todos sus datos por pantalla.
- 9.- Eliminar un producto => Debe solicitar el producto a eliminar. Una vez localizado se procederá a eliminarlo del fichero.

Notas.

- Todas las opciones del menú que se muestren datos tendrán que indicar qué son, por ejemplo, si se ha de mostrar el nombre y la cantidad de un producto aparecerá en pantalla:

Producto: Naranjas, Cantidad: 10

- En el caso de mostrar listados, éstos tendrán una cabecera donde la primera línea contenga el nombre y la dirección del almacén y en la segunda línea de la cabecera los datos que se van a mostrar de manera tabular. Por ejemplo:

```
Listado del almacén "Productos el Paso" localizado en Avenida Tres Cuartos SN
*****
CODIGO  NOMBRE                                PRECIO  CANTIDAD  FECHA CADUCIDAD
CAPc10  Aceite de oliva                        8.21429    61        12/12/2032
CAPc99  Aceite de oliva virgen extra          11.7       118       29/12/2032
```

Clases, tipos de datos y formatos de ficheros**Fichero GestorAlmacenes.x**

```
typedef char Cadena[90];

struct TFecha
{
    int Dia;
    int Mes;
    int Anyo;
};

struct TProducto
{
    Cadena CodProd;
    int Cantidad;
    Cadena NombreProd;
    float Precio;
    Cadena Descripcion;
    TFecha Caducidad;
};
```

```

struct TDatosAlmacen
{
    Cadena Nombre;
    Cadena Direccion;
    Cadena Fichero;
};

struct TBusProd
{
    int Almacen;
    Cadena CodProducto;
};

struct TObtProd
{
    int Almacen;
    int PosProducto;
};

struct TActProd
{
    int Almacen;
    TProducto Producto;
};

struct TOperacion
{
    int p1;
    Cadena p2;
    float p3;
    float p4;
};

program SUPERMERCADO {
    version SUPERMERCADO_VER {

        //Dado una posición del almacén, devuelve sus datos
        TDatosAlmacen DatosAlmacen(int pAlmacen)=1;

        //Dado una posición de almacén, devuelve el número de productos o -1 si no existe.
        int NProductos(int pAlmacen)=2;

        //Crea un almacén vacío de productos y devuelve la posición donde se encuentra. Si previamente ya
        //estaba creado y cargado en memoria, devolverá la posición donde se encuentra en memoria.
        int CrearAlmacen(TDatosAlmacen pDatos)=3;

        //Abre un fichero de almacén y lo carga en memoria y devuelve su posición. Si previamente ya estaba
        //cargado devuelve la posición donde se encuentra.
        int AbrirAlmacen(Cadena pNomFiche)=4;

        //Dado la posición de un almacén vuelca sus datos al fichero. Devuelve true si ha podido guardarlo.
        bool GuardarAlmacen(int pAlmacen)=5;

        //Dado la posición de un almacén, vuelca sus datos al fichero y elimina la memoria dinámica asociada.
        //Si el Almacén es compartido, solo el último cliente es el que eliminará la memoria dinámica.
        bool CerrarAlmacen(int pAlmacen)=6;

        //Dado la posición del almacén devuelve true si el almacén está cargado en memoria.
        bool AlmacenAbierto(int pAlmacen)=7;

        //Dado la posición del almacén y un código de producto, devuelve la posición dentro del vector de
        //dinámico donde se encuentra el producto. Si no lo encuentra devuelve -1.
        int BuscaProducto(TBusProd pBus)=8;

        //Dado la posición del almacén y la del producto, lo devuelve.
        TProducto ObtenerProducto(TObtProd pPos)=9;

        //Dado la posición de un almacén y un producto lo añadirá a la memoria dinámica del almacén y devuelve
        //true. No puede existir dos productos con el mismo código en el almacén.
        bool AnadirProducto(TActProd pProdNuevo)=10;

        //Dado la posición de un almacén y un producto lo actualizará en la memoria dinámica del almacén.
        //Devolverá true si el producto existe y ha sido actualizado.
        bool ActualizarProducto(TActProd pProd)=11;
    }
}

```

```
//Dado la posición del almacén y un código de producto, lo elimina de la memoria dinámica y devuelve
//true.
bool EliminarProducto(TBusProd pProduc)=12;

//No hay que implementarlo por ahora
TProducto operacion(TOperacion)=13;

    } = 1;
} = 0x30000001;
```

Estructura de Datos para almacenar en memoria dinámica los almacenes

```
typedef struct {
    Cadena NombreA,DireccionA, NombreFile;
    TProducto *Productos;    //Vector dinámico
    int NProduc;             //Número de productos
    int NClientes;           //Número de clientes que comparten el almacén.
} TAlmacen;

TAlmacen *Almacenes=NULL;   //Vector dinámico de Almacenes
int NAlmacenes=0            //Número de Almacenes.

//Notas
Cuando un almacén es eliminado (no hay ningún cliente trabajando con él) los datos del vector se inicializan
de manera que la misma posición del vector puede ser utilizado cuando se cree un nuevo almacén o bien se
abra algún fichero de almacén.
```

Ficheros de la aplicación

Para esta práctica, se proporciona un fichero de almacén (Almacen.dat) con datos para ayudar a la codificación y la comprobación del código.

El fichero de almacén tiene la siguiente estructura:

Cabecera del Fichero			Productos del Almacén			
Número de Productos	Nombre Almacén	Dirección Almacén	Producto 1	Producto 2	...	Producto N

Dispone una cabecera formada por el número de productos almacenados, el nombre y la dirección del almacén. A continuación de la cabecera estarán el conjunto de productos del almacén.

Un ejemplo del contenido parcial del fichero Almacen.dat es:

Almacen.dat

```
Listado del almacén "Productos el Paso" localizado en Avenida Tres Cuartos SN
*****
CODIGO  NOMBRE                PRECIO  CANTIDAD  FECHA CADUCIDAD
CAPc10  Aceite de oliva           8.21429 61        12/12/2032
CAPc99  Aceite de oliva virgen extra 11.7    118       29/12/2032
CAPc40  Aceite de coco            11.8    124       4/10/2031
CMPo91  Mostaza                   8.76923 81        10/2/2032
CMPo50  Mostaza en grano          13.0909 64        8/6/2033
CCPu56  Curry amarillo            18.5    131       29/6/2032
CCPu84  Curry rojo en pasta       6.88889 103       23/12/2032
CGPu29  Guacamole                 19.1429 73        6/8/2031
CTPo88  Tomate frito              7.57143 62        16/11/2031
CTPo94  Tomate natural            8.69231 90        2/8/2032
CTPo40  Tomate rallado            11.5455 85        9/10/2033
```

Realización de la práctica en grupo

Esta práctica se puede realizar de manera individual o por grupos. El número de alumnos en cada grupo es dos. Los alumnos que decidan hacerla en grupo deberán notificar al profesor los componentes del grupo y el reparto equitativo de tareas que cada alumno va a realizar. El profesor podrá reorganizar las tareas en caso de que detecte un reparto desigual.