



1. Se desea implementar una pequeña aplicación que permita almacenar información sobre recursos (documentos, programas, aplicaciones, imágenes, ficheros de audio, etc.) que un usuario descarga de Internet. La información que se necesita manejar básicamente es:
  - Servidor web. Lugar web (web site) donde reside el recurso que se ha descargado. Básicamente se desea información del servidor, el protocolo y el puerto de comunicaciones. Por ejemplo: `http://miservidor.midominio.es:8080` es una URL donde el protocolo es el `http`, el servidor es `miservidor.midominio.es` y el puerto es el `8080`.
  - Recurso descargado. Información sobre el recurso que se ha descargado, que consiste básicamente en el nombre del fichero descargado, una descripción del mismo y una categoría.

Se pretende que toda esta información quede almacenada en ficheros de forma no redundante. Esto significa que la aplicación trabajará con 2 ficheros de texto independientes, de modo que uno contendrá información de los servidores web (llámalo `web.txt`), otro se destinará a información sobre los recursos (llámalo `recursos.txt`).

La forma habitual de trabajo con ficheros consiste en cargar en memoria los datos de los ficheros (por ejemplo, en estructuras de tipo vector) con el objetivo de que el acceso a la información y su modificación sea lo más rápida y cómoda posible. Por este motivo, **el programa deberá funcionar de modo que al inicio del mismo se carguen los datos en memoria, utilizando las estructuras adecuadas (ver la sección de ayuda al final del enunciado), y los datos que se han trabajado en memoria se guarden en los ficheros al finalizar la ejecución del programa, previa validación del usuario (el usuario decide si al terminar los cambios se guardan o no en los ficheros).**

El funcionamiento de la misma estará basado en un menú con las siguientes opciones:

1. Listar servidores
2. Listar recursos
3. Buscar recurso
  
4. Insertar servidores
5. Insertar recursos
  
6. Modificar/eliminar servidores
7. Modificar/eliminar recursos
  
8. Guardar datos en ficheros
  
0. Salir

El proceso de trabajo de la aplicación se basará en cargar en memoria los datos de los ficheros **al iniciar la aplicación** para obtener mayor rendimiento en las operaciones, de forma que se



utilizarán vectores de estructuras para almacenar los datos de los ficheros (con un límite máximo de 1000 elementos del vector).

Si no existen los ficheros inicialmente, éstos se crearán utilizando las opciones 3 y 4 del programa y guardándolos posteriormente con la opción 7.

Al finalizar la aplicación, se solicitará al usuario si desea guardar los cambios realizados en los ficheros, y se obrará en consecuencia. También será posible guardar los datos en los ficheros como una opción determinada del menú.

Las **restricciones** y requisitos de la aplicación son los siguientes:

- En los listados se debe mostrar el conjunto completo de registros que corresponden con cada categoría (servidores, recursos).
- No puede existir un recurso que haga referencia a un servidor que no existe en el sistema.
- Las modificaciones permiten cambiar características de servidores y recursos (sin permitir duplicados).
- Las eliminaciones se pueden gestionar en memoria de modo que los registros que se pretendan borrar queden "marcados" en memoria, y sólo cuando se haga un volcado de registros a ficheros se borrarán físicamente (es decir, no se copiarán de memoria a fichero). Cuando se elimina un servidor deben también marcarse/eliminarse los recursos que contiene.
- No se podrán insertar servidores duplicados, aunque sí se permite que un recurso esté disponible en varios servidores.
- La opción de buscar recurso sirve para localizar un recurso, de modo que debe buscar el vector de recursos a través de los campos nombre/descripción y listar los resultados coincidentes.

## AYUDA

Para facilitar las tareas de programación, se recomiendan las estructuras de datos y formatos siguientes. (El alumno puede plantear sus estructuras y formatos adicionales si lo cree conveniente o si son más adecuados para la lógica de su aplicación.)

### **Estructuras de datos recomendadas**

```
SERVIDOR {  
REF_SERVIDOR      ENTERO,  
PROTOCOLO        CADENA,  
PUERTO           ENTERO,  
DIRECCION        CADENA,  
DESCRIPCION      CADENA
```



}

RECURSO {

REF_SERVIDOR	ENTERO
REF_RECURSO	ENTERO,
NOMBRE	CADENA,
CATEGORIA	CADENA,
DESCRIPCION	CADENA

}