

Se pide desarrollar un servicio - llamado *gatoRemoto* - que permita visualizar, de forma remota, el contenido de ficheros de texto. Básicamente, el servicio recibe el nombre de un fichero y devuelve el contenido del mismo. Para implementar este servicio se hará uso de Servicios Web. Tanto la parte cliente, como la parte servidor, estarán escritos en C utilizando gSoap.

La siguiente porción de código muestra el contenido del fichero `wsRemoteCat.h`

```
//gsoap gatoRemoto_ns service name: gatoRemoto
//gsoap gatoRemoto_ns service style: rpc
//gsoap gatoRemoto_ns service encoding: encoded
//gsoap gatoRemoto_ns service namespace: urn:remoteCat

/** Length of data buffer */
#define MAX_DATA_SIZE 1024

/** Length for tString */
#define NAME_LENGTH 128

/** Typedef for the file name */
typedef char *xsd_string;

/** Struct that contains a file name */
typedef struct tFileName{
    int __size;
    xsd_string __ptr;
}gatoRemoto_ns__tFileName;

/** Binary data */
typedef struct tData{
    int __size;
    unsigned char *__ptr;
}gatoRemoto_ns__tData;

int gatoRemoto_ns__execGatoRemoto (gatoRemoto_ns__tFileName fileName,
                                   gatoRemoto_ns__tData *data);
```

La estructura `gatoRemoto_ns__tFileName` se utilizará para indicar el nombre de los ficheros. La longitud del nombre de los ficheros puede ser variable, pero en ningún caso superará los `NAME_LENGTH` caracteres. De forma similar, la estructura `gatoRemoto_ns__tData` representa el contenido de los ficheros, pudiendo almacenar hasta `MAX_DATA_SIZE` caracteres.

El servicio pedido se representa mediante la función `gatoRemoto_ns__execGatoRemoto`, la cual recibe el nombre de un fichero en el parámetro `fileName` y, como salida, devuelve su contenido en la estructura `data`.

Consideraciones de implementación:

- Los ficheros en el servidor contienen, como máximo, `MAX_DATA_SIZE` caracteres.
 - Cada operación de lectura se realiza completamente con una invocación.
 - Si el fichero indicado por el cliente no existe en el servidor, se devolverá la estructura vacía.
- a) Implementad la parte cliente para que utilice el servicio descrito. La invocación se realizará de la siguiente forma:

```
./client http://server:port sourceFile
```

donde `server` es la dirección del servidor, `port` es el puerto donde escucha peticiones el servidor y `sourceFile` es el nombre del fichero.

- b) Implementad la parte servidor que contenga el servicio descrito en el fichero `wsRemoteCat.h`. Su invocación se realizará de la siguiente forma:

```
./server port
```

donde `port` es el puerto donde el servidor escuchará las peticiones de los clientes.

Para facilitar la implementación del servicio, se puede hacer uso de la siguiente función, la cual se proporciona ya implementada:

```
int calculateFileSize (char* fileName);
```

Esta función recibe como parámetro el nombre de un fichero y devuelve su tamaño, en bytes. En caso de que el fichero no exista, devuelve -1.

- c) Describid los pasos para desplegar el servicio. No es necesario indicar el nombre exacto de cada fichero generado, pero sí la funcionalidad que desempeñan.