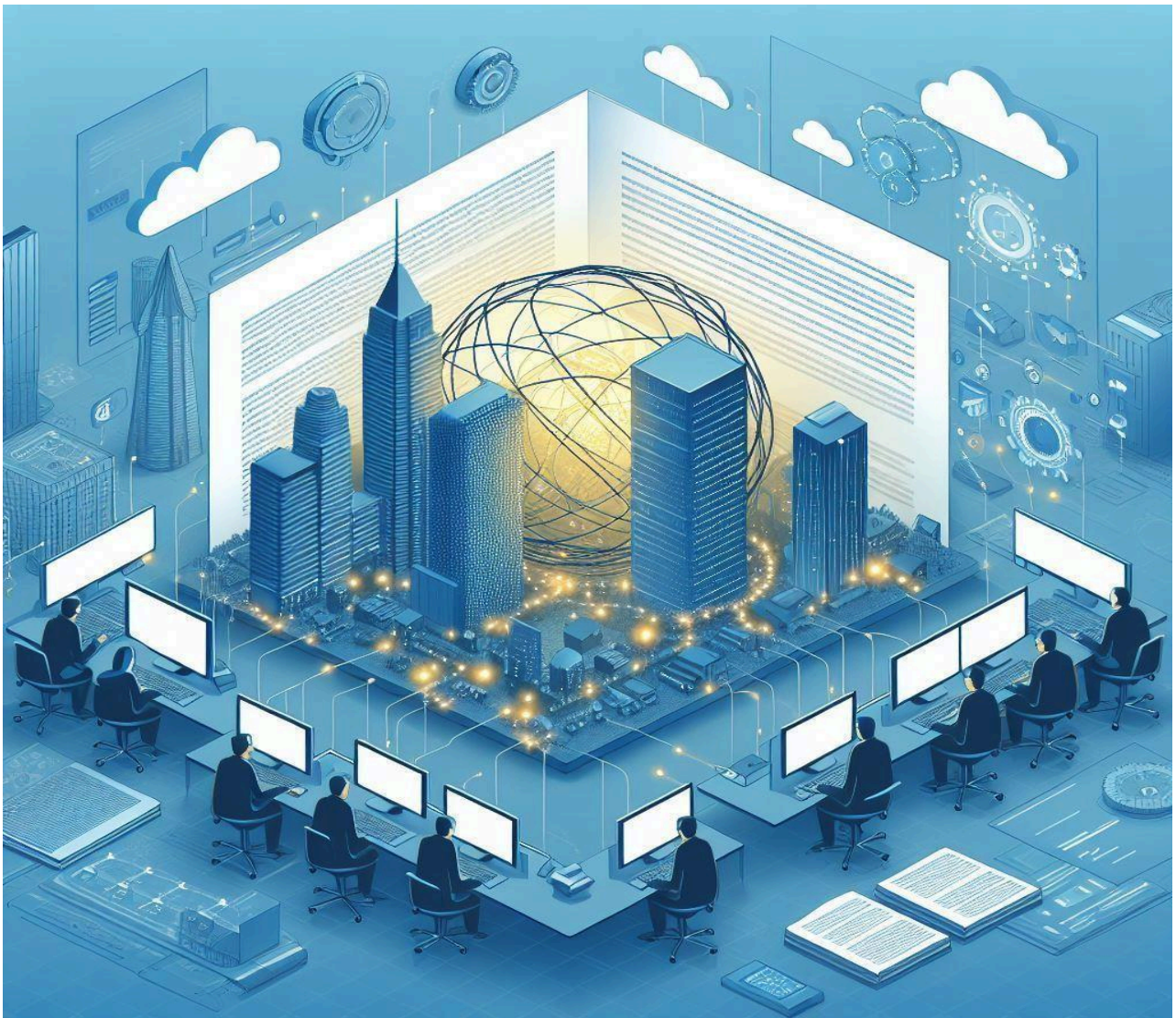

DOCUMENTACIÓN DE DISEÑO

Redes de la Comunicación



Índice

Índice	1
1. Formato de los mensajes del protocolo de comunicación con el Directorio	2
2. Formato de los mensajes del protocolo de transferencia de ficheros	6

1. Formato de los mensajes del protocolo de comunicación con el Directorio

Para definir el protocolo de comunicación con el *Directorio*, vamos a utilizar mensajes textuales con formato “campo:valor”. El valor que tome el campo “operation” (código de operación) indicará el tipo de mensaje y por tanto su formato (qué campos vienen a continuación).

Cada petición realizada por el cliente de NanoFiles al directorio vendrá acompañado de la respuesta, por parte del directorio, con el éxito o fracaso de la petición.

Tipos y descripción de los mensajes

Mensaje: **login**

Parámetros: La operación ‘login’ dirección del directorio donde se quiere dar de alta y el nombre del nickname.

Sentido de la comunicación: Cliente→ Directorio

Descripción: Este mensaje lo envía el cliente de NanoFiles al Directorio para solicitar “iniciar sesión” y registrar el nickname indicado en el mensaje.

Ejemplo:

```
operation: login\n
nickname: alicia\n
\n
```

Mensaje: **loginOk**

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente NanoFiles informándole del éxito de la operación de inicio de sesión y proporcionándole la clave de la sesión (sessionKey), la cuál será necesaria para todas las siguientes peticiones del cliente.

Ejemplo:

```
operation: loginOk\n
sessionkey: 787\n
\n
```

Mensaje: loginFail

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente NanoFiles informándole del fracaso de la operación de inicio de sesión (nickname proporcionado ya en uso).

Ejemplo:

```
operation:loginFail\n\n
```

Mensaje: logout

Parámetros: Simplemente la operación 'logout', el software se encarga de proporcionar la clave de sesión al directorio.

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje lo envía el Cliente al Directorio para solicitar el cierre de sesión y borrar su nickname del directorio. Se envía la clave de sesión para que el directorio compruebe que es un cliente cuyo nickname está registrado.

Ejemplo:

```
operation:logout\nsessionkey:787\n\n
```

Mensaje: logoutOk

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente NanoFiles informándole del éxito de la operación de cierre de sesión. Tanto su nickname como su sessionkey han sido dados de baja.

Ejemplo:

```
operation:logoutOk\n\n
```

Mensaje: logoutFail

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente NanoFiles informándole del fracaso de la operación de cierre de sesión. Tanto su nickname como su sessionkey han sido dados de baja.

Ejemplo:

```
operation:logoutFail\n\n
```

Mensaje: `getUserList`

Parámetros: Simplemente la operación 'userlist', una vez más el software es el encargado de proporcionar la clave de sesión al directorio.

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje lo envía el cliente al directorio como petición para obtener una lista de todos los usuarios conectados al directorio en ese momento. Se envía la clave de sesión para que el directorio compruebe que es un cliente cuyo nickname está registrado.

Ejemplo:

```
operation:getUserList\n
sessionkey:787\n
\n
```

Mensaje: `getUserListOk`

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente para informarle del éxito al obtener la lista de usuarios, además de pasarle la misma en formato String. El directoryconnector se encarga de imprimir por pantalla la lista entre corchetes.

Ejemplo:

```
operation:getUserListOk\n
userlist:juan,jose\n
\n
```

Mensaje: `getUserListFail`

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente para informarle del fracaso al obtener la lista de usuarios.

Ejemplo:

```
operation:getUserListFail\n
\n
```

Mensaje: `registerServer`

Parámetros: Cuando se registra un cliente para a ser un servidor de ficheros, se tiene que dar de alta como tal en el directorio, es decir tecleando 'fgserve' o 'bgserve'. El software es el encargado de proporcionar la clave de sesión al directorio.

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje se envía desde el cliente al directorio y se envía cada vez que se un cliente pasa a ser un servidor de ficheros, tanto el primer como en segundo plano. Es una petición para darse de alta como servidor en el directorio. El mensaje contiene el puerto donde este fichero esta escuchando las peticiones de otros clientes.

Ejemplo:

```
operation:registerServer\n
sessionKey:787\n
serverPort:57842\n
\n
```

Mensaje: registerServerOk

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente para informar del éxito de la operación de registro de servidor.

Ejemplo:

```
operation:registerServerOk\n
\n
```

Mensaje: registerServerFail

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente para informar del fracaso de la operación de registro de servidor.

Ejemplo:

```
operation:registerServerFail\n
\n
```

Mensaje: deregisterServer

Parámetros: Cuando un servidor de ficheros deja de serlo tiene que darse de baja en el directorio como servidor, es decir, tecleando 'stopserver' o 'fgstop'. El software envía la clave de sesión.

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje es enviado del cliente al directorio como petición para darse de baja como servidor en el directorio. Quita el puerto donde este servidor escuchaba las peticiones de otros clientes.

Ejemplo:

```
operation:deregisterServer\n
sessionkey:787\n
\n
```

Mensaje: deregisterServerOk

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente para informar del éxito de la operación para darse de baja como servidor.

Ejemplo:

```
operation:deregisterServerOk\n
\n
```

Mensaje: deregisterServerFail

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje lo envía el directorio al cliente para informar del fracaso de la operación para darse de baja como servidor.

Ejemplo:

```
operation:registerServerFail\n\n
```

Mensaje: **lookupserveraddr**

Parámetros: Este mensaje se envía cuando en el comando ‘downloadfrom’ hace referencia al servidor de ficheros a través de su nickname y no por su ip:puerto.

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje lo envía el cliente al directorio para solicitar la dirección de cierto servidor de ficheros (por su nickname). El directorio comprueba si ese usuario esta dado de alta como servidor y si es así devuelve su dirección.

Ejemplo:

```
operation:lookupserveraddr\nsessionkey:787\nservername:juan\n\n
```

Mensaje: **lookupserveraddrOk**

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje se envía para informar del éxito de la operación para obtener la dirección de un servidor de ficheros. Devuelve el puerto donde escucha y la ip de si host.

Ejemplo:

```
operation:lookupserveraddrOk\nserverport:53542\nserverip:154.54.5.206\n\n
```

Mensaje: **lookupserveraddrFail**

Sentido de la comunicación: Directorio -> Cliente

Descripción: Este mensaje se envía para informar del fracaso de la operación para obtener la dirección de un servidor de ficheros (el host proporcionado no es un servidor de ficheros).

Ejemplo:

```
operation:lookupserveraddrFail\n\n
```

2. Formato de los mensajes del protocolo de transferencia de ficheros

Para definir el protocolo de comunicación con un servidor de ficheros, vamos a utilizar mensajes binarios multiformato. El valor que tome el campo “opcode” (código de operación) indicará el tipo de mensaje y por tanto cuál es su formato, es decir, qué campos vienen a continuación.

Tipos y descripción de los mensajes

Mensaje: **FileNotFound (opcode = -1)**

Sentido de la comunicación: Servidor de ficheros → Cliente

Descripción: Este mensaje lo envía el par servidor de ficheros al par cliente (receptor) de fichero para indicar que no es posible encontrar el fichero con la información proporcionada en el mensaje de petición de descarga.

Ejemplo:

Opcode (1 byte)
-1

Mensaje: **downloadfrom (opcode = 1)**

Sentido de la comunicación: Cliente -> Servidor de ficheros

Descripción: Este mensaje lo envía el par cliente al par servidor para la petición de una descarga de un archivo. Este mensaje se envía cuando el usuario teclea el comando 'downloadfrom':

downloadfrom <nameServer> <fileHash> <nameLocalFile>'. De estos datos los que nos interesan a nosotros es el hash. Encontrar la dirección del servidor de ficheros es responsabilidad del controlador del cliente-servidor. Por tanto, para enviar el hash debemos crear dos campos, uno con la longitud del hash y otro con el propio hash. Saber la longitud es necesario para poder utilizar el método readFully() en la clase PeerMessage.

Ejemplo:

Opcode (byte)	hashLen (integer)	codeHash (String)
1	4	"b4q3"

Mensaje: **downloadfromOk (opcode = 2)**

Sentido de la comunicación: Servidor de ficheros -> Cliente

Descripción: Este mensaje es la respuesta, por parte de servidor de ficheros, al anterior mensaje. Es la confirmación de que la petición se ha satisfecho correctamente, es decir, que el hash proporcionado corresponde estrictamente a un solo fichero (de entre todos los ficheros de NF-SHARED). Dicha confirmación manda el hash completo del fichero en cuestión, ya que se puede pasar una subcadena del hash completo. Necesitamos el hash completo porque el cliente tiene que hacer una última comprobación para asegurarse que el fichero descargado es el deseado: comparamos el hash proporcionado por el servidor de ficheros con el hash que devuelve el método auxiliar FileDigest.computeFileChecksumString(). De igual manera que en el mensaje anterior, tenemos que pasar la longitud del hash para poder utilizar readFully().

Ejemplo:

Opcode (byte)	hashLen (integer)	codeHash (String)
2	40	"dwd3f3 . . . fw4fwf"

Mensaje: **downloadfromFail (opcode = 3)**

Sentido de la comunicación: Servidor de ficheros -> Cliente

Descripción: Este mensaje es la respuesta, por parte de servidor de ficheros, a la petición de descarga. Es la confirmación de que la petición no ha salido correctamente, es decir, que el hash proporcionado es ambiguo o no corresponde a ningún fichero (de entre todos los ficheros de NF-SHARED). Dicha confirmación manda un código de fallo: hash ambiguo (0) o hash sin coincidencias (1). Es responsabilidad del cliente interpretar el mensaje e informar al usuario de la razón de fallo.

Ejemplo:

Opcode (byte)	codeFail (byte)
3	0 ó 1

Mensaje: **content (opcode = 4)**

Sentido de la comunicación: Servidor de ficheros -> Cliente

Descripción: Este mensaje es el encargado de enviar el contenido del fichero. Como mandamos el contenido por partes será enviado varias veces en la misma descarga. El funcionamiento es el siguiente: el cliente solicita la descarga; el servidor comprueba que solo un archivo coincide con el hash; el servidor envía la confirmación de que la descarga se va a hacer; el servidor envía tantos mensajes de tipo 'content' como sean necesarios para enviar todos los bytes del archivo. Los campos del mensaje son los siguientes: opcode; índice (para saber cuál es el último bit enviado); la longitud del contenido total del fichero; la longitud del array de bytes que estamos enviando (siempre va a tener longitud 40000, menos el último mensaje que tendrá la longitud de los bytes que quedan por enviar); el array de bytes del contenido.

Ejemplo:

Opcod e (byte)	Indice (long)	contenidoTotalLen (long)	dataLe n (long)	Conten t (byte[])
4	40000	163244	40000	37,80,35,...,34