Informe Tarea 3
Sistemas Operativos y Redes

Gianfranco Traverso Katherine Jara

Noviembre 2019

Nuestro protocolo estará inspirado en el famoso protocolo de HTTP debido ha que es un de los mas populares y simples de implementar. El nombre que le pondremos al protocolo será GK.

1 Tipos de mensaje.

Un mensaje es el medio por el cual se intercambian datos entre clientes y servidores. Estos mensajes siguen una estructura pre-definida para entregar información al receptor y así poder determinar una acción. Los mensajes se pueden dividir en dos categorías, Peticiones y Respuesta.

1.1 Petición.

Las peticiones son exclusivas del cliente. Sirven para pedirle al servidor distintas acciones que el cliente puede necesitar. La petición se puede dividir en 3 secciones:

1.1.1 Línea de inicio:

Mensaje realizado por un cliente con el que comienza la acción con el servidor. Está formada por tres elementos:

a) Un método , un nombre o un verbo: GET - LIST - PEEK que especifiquen la acción que se quiere realizar.

b) El objetivo de la petición, usualmente se utiliza una URL o la dirección completa del protocolo, puerto y dominio. Este formato puede variar según los métodos implementado.

c) La versión del portocolo, este define la estructura de los mensajes (indica la versión que se espera para la respuesta).

1.1.2 Cabeceras:

Está compuesta por una serie de caracteres (no diferencia mayúsculas ni minúsculas), luego dos puntos ":" y finalmente un valor. Se pueden clasificar en: **generales** quienes afectan al mensaje completo. **Petición** como: Accept-Type, User-Agent, modifican la petición entrando más en detalle (Accept-Languaje), restringuiéndola condicionalmente (IF-None) o entregándole un contecto (Referer). **Entidad** quienes se aplican al cuerpo de la petición (Content-Lenght).

1.1.3 Cuerpo:

Existen cuerpos con un **único dato**, el cual se basa en un archivo definido por las cabeceras *Content-Type* y *Conent-Lenght*. Y cuerpos con **múltiples datos**, está compuesto por diversos contenidos (formularios HTML).

1.2 Respuesta

La respuestas, como lo sugiere su nombre, es el mensaje en respuesta a la petición del cliente que está interactuando con el servidor. La respuesta del protocolo se puede dividir en tres secciones, la **Linea de estado**, **Cabeceras**, **Cuerpo**.

La **Linea de estado** contiene la versión del protocolo, un código de estado que indica si la petición fue exitosa o rechazada y finalmente un pequeño texto que tiene el objetivo de informar sobre el tipo del código de estado.

La **Cabecera** contiene una cadena de texto seguido por dos puntos "**:**" y un valor que dependerá del tipo de cabecera que se esté usando. Existen muchos tipos de cabeceras, en donde se pueden agrupar usando los mismos tipos de cabeceras mencionadas en los mensajes de petición.

El **Cuerpo** corresponde a la última parte del mensaje. Al igual que las cabeceras, el cuerpo también se puede categorizar en distintos tipos, que corresponden a las mismas categorías mencionadas en la sección de petición.

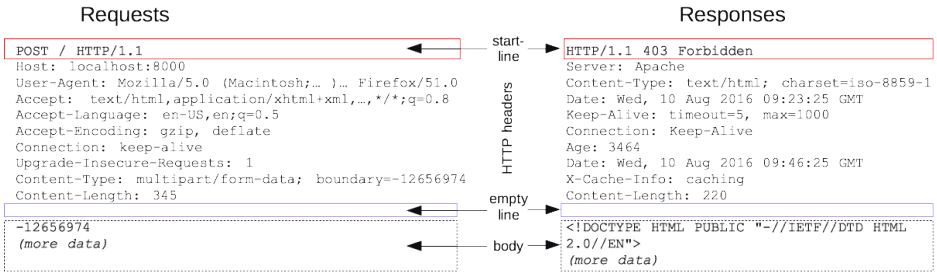

Figure 1: Ejemplo de petición y respuesta sacado del protocolo http

2 Acciones de GK

Las acciones o métodos de nuestro protocolo son una serie de instrucciones que ya vienen pre-definidas al implementar el protocolo. Estás instrucciones tienen la finalidad de ejecutar acciones básicas que facilitan la comunicación e interacción entre el cliente y servidor. A continuación veremos cada uno de estos métodos con un breve resumen respecto a su funcionalidad.

GET: Este metodo entrega al servidor un *value*, en caso de encontrarse en la base de datos, el servidor retornará la *key* del *value* solicitado.

LIST: Este método le pide al servidor una lista con todos los datos guardados y sus respectivas *key*.

OPTION: Este método representa una petición por información sobre las distintas opciones de comunicación o métodos disponible en los mensajes de petición y respuesta que soporta el protocolo.

INSERT: Se usa para crear un nuevo dato en el servidor. Para este metodo, se puede especificar una *key* para el valor que se desea agregar, en caso de que la *key* ya exista, el servidor no actualizará el dato, si no se especifique ninguna, el servidor creará una *key* aleatoria no existente.

UPDATE: Este método le entrega al servidor una *key* y un *value* con la finalidad de actualizar el valor que este asociado con esa llave y remplazarlo por el nuevo valor entregado.

DELETE: Borra el dato especificado en el servidor, para referenciar el dato, es necesario especificar su *key*.

CONNECT: Este método se usa para establecer una conexión con el servidor.

DISCONNECT: Permite desconectarse del servidor de origen, cerrando el socket entre ambos programas.

PEEK: Este método sirve para verificar la existencia de una key que esté en la base de datos, retorna *True* en caso de que exista y *False* en el caso contrario.

.

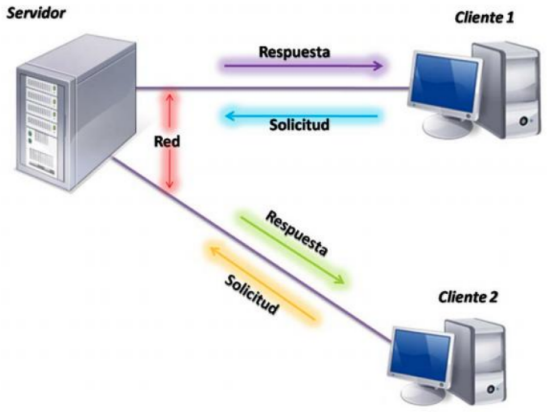

Figure 2: Interacción posible del protocolo

3 Estados de protocolo

El estado será los caracteres GK más números que irán en el header (GKXX)

- GK10 OK (mensaje enviado correctamente, sin errores)
- GK20 ERROR CONNECT (tiempo expira y el cliente no puede conectarse al servidor)
- GK40 NOT FOUND (cuando ingresa un comando que no existe)
- GK50 KEY NOT FOUND (cuando intenta actualizar o eliminar una key que no existe en la base de datos)
- GK60 ERROR INSERT (cuando intenta ingresar una key que ya existe en la base de datos)
- GK70 ERROR DISCCONECT (sucede algún inconveniente y no es posible desconectarse del servidor)
- GK80 ERROR DELETE (no es posible eliminar la key)