

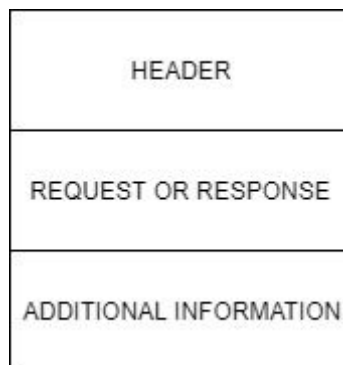
Informe Tarea 3 Parte 1

Nombres: Andrés Vial - Matías Ponce

Protocolo JOJO

Definición de tipos de mensaje en el protocolo:

El tipo de mensaje que usa nuestro protocolo es similar al de DNS, en el sentido de que los *request* de los clientes y los *response* del servidor tienen el mismo formato, el cual es así:



Header:

ID/IP/ROR/CODE/ESTADO/

- ID: Identificador que debe coincidir entre los request y response, para por ejemplo, el servidor no mande un mensaje al cliente equivocado.
- IP: La dirección del dispositivo que manda el mensaje
- ROR: Bit que define si el mensaje es request o response (0 para request y 1 para response)
- CODE: Código del mensaje
- ESTADO: Indica el estado del emisor

Request:

COMMAND/ADD/KEY/VALUE/

- COMMAND: Comando que se puede utilizar en la base de datos
- ADD: Bit que identifica si se necesita información adicional (0 si no necesita, 1 si necesita)
- KEY: Bit que identifica si el comando necesita un key para realizar el comando (0 si no necesita, 1 si necesita)
- VALUE: Bit que identifica si el comando necesita un value para realizar el comando (0 si no necesita, 1 si necesita)

Response:

RESPUESTA/

- RESPUESTA: string con la respuesta de la petición realizada

Additional Information:

Contendrá la key y el valor adicional si se necesita, ya sea en comandos como insert(key,value)

KEY/VALUE/

- KEY: Contendrá un texto con la Key
- VALUE: Contendrá un texto con el value

Ejemplo del formato del mensaje:

Request:

ID/IP/ROR/CODE/ESTADO	->	10/172.16.4.205/0/dhasfe/100
COMMAND/ADD/KEY/VALUE/	->	insert/1/1/1/
KEY/VALUE/	->	1234/hola/

Response:

ID/IP/ROR/CODE/ESTADO	->	10/172.16.1.1/1/jasjrja/500
RESPUESTA/	->	"Inserción Correcta"

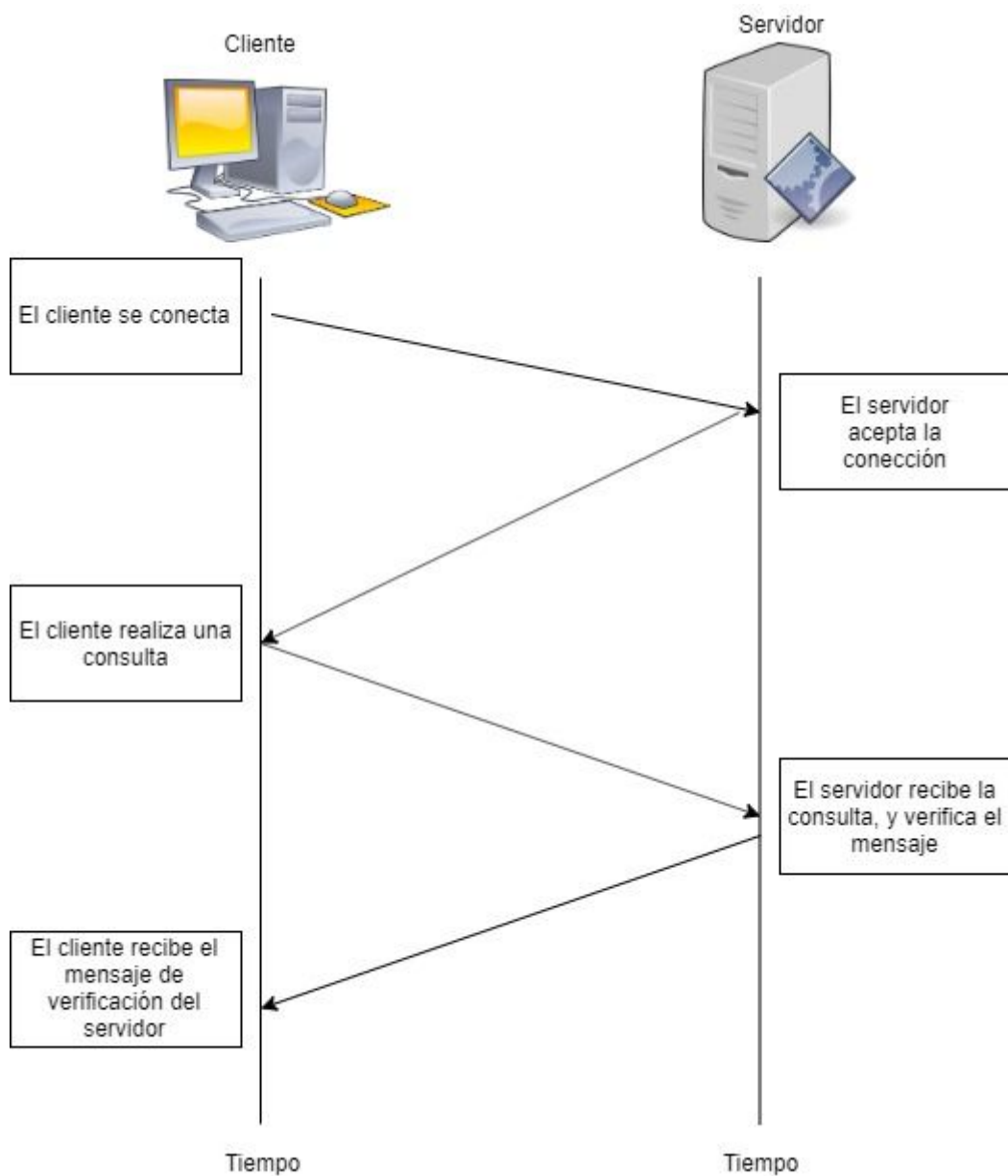
Definición de acciones del protocolo:

Las acciones que se pueden realizar con los COMMAND que van dentro del header, estos son:

- CONNECT: Comando para conectarse al servidor
- DISCONNECT: Comando que permite al cliente desconectarse del servidor.
- QUIT: Comando que cierra la conexión actual (si la hay), y cierra el cliente.
- INSERT: Comando que inserta una nueva tupla clave-valor en el servidor. Se genera error si la llave ya existe en la base de datos. Si no se proporciona una clave, el servidor generará una

- GET: Comando que retorna el valor asociado a la clave ingresada
- PEEK: Comando que retorna 'true' si la clave está registrada en la base de datos, o 'false' si no lo está.
- UPDATE: Comando que permite actualizar el valor asociado a la clave ingresada. Retorna error si la clave ya está en la base de datos.
- DELETE: Comando que permite eliminar un par clave-valor con la clave ingresada.
- LIST: Comando que retorna un listado con todas las claves disponibles.

El diagrama de transición se vería así:



Definición de estados del protocolo:

El estado será un número que irá dentro del header, el cual hará referencia a los siguientes estados que podrán ocurrir:

0 = DISCONNECT
50 = CONNECT
100 = ENVIADO (Sólo para los mensajes que envíe el cliente)
150 = ERROR INSERT
200 = ERROR GET
250 = ERROR PEEK
300 = ERROR UPDATE
350 = ERROR DELETE
400 = ERROR LIST
500 = OK (El mensaje se recibió bien y no hubo error)

Cuando el estado sea uno de los errores, el error irá detallado en la parte de “información adicional” del mensaje. Por ejemplo: Si ocurriera el error 200 porque no existe la key ingresada, en la sección de “información adicional” diría algo como “Error en get, no existe la clave ingresada en la base de datos”.