

# Informe Tarea 3 parte 1

Carlos Águila Kiblsky

Mensajes dentro del protocolo tendrán el siguiente formato:

## Header-Content

**Header**(común para request, response)

ID: este se utiliza para corroborar que el servidor se este comunicando con el cliente correcto y viceversa (Int)

IP: Dirección de donde se envía el mensaje, con este dato el servidor vera de donde vienen sus mensajes y el cliente observara de donde es el servidor. (Int)

Modo: este es in bit que puede ser 0 o 1 que indica si es request(0) o response(1)

Estado: posibilidad que puede tener un mensaje (Int)

## Contenido:

**Response**:(después del procesamiento de la petición entrega lo siguiente de acuerdo a la petición)

Petición	tipo respuesta
Insert	Bool
Get	string
Delete	Bool
Update	Bool
List	String
Peek	Bool

**Request**(ver tabla): **Comand**(descritos en el enunciado)

**Content\_type**

**Body**(type del content type en orden )

Comand	Content-type	Body
--------	--------------	------

Insert	Key-value Value	Int-string Int-Int Int String
Get	Key	Int
Delete	Key-Key	Int-Int
Update	Key-value	Int – string Int – Int
DisConect		
Conect		
Quit		
List		
Peek	Key	Int

Usando un sistema parecido al de DSN logre simplificar la mensajería y parametrizarla sin perder certeza de lo que se manda y lo que se recibe. Los host, dentro de este protocolo, tendrán que corroborar datos del header antes de pasar al contenido.

Los host que reciban respuesta, tendrán que recibirla con los parámetros que esperan en el contenido de la respuesta, así verificaran aun mas, si es la respuesta que desean obtener.

Dentro de las respuestas obtenidas se pueden encontrar los siguientes estados:

0	Disconnect
1	Conect
50	Enviado(Cliente)
100	Error de Update
200	Error de Delete
300	Error de Get
400	Error de Insert
500	Error de Peek
600	Error de List
666	Error desconocido
999	Todo en orden(servidor-cliente)

Claro esta que el contenido del header es el que define como se va a interpretar el contenido del mensaje en los host del sistema.

Diagrama:

Cliente	Medio	Servidor
		Escucha

Petición de Conexión		
		Acepta conexión
Request		
		Verify después Response
Verify Output		