|  |
| --- |
| Manual_final_image001  **facultad de ciencias empresariales** |
| Informe Tarea 1 |
| Comunicación de Datos y Redes |

Alumno.

Nicolás Oyarce Aburto

Profesor.

Patricio Galdames Sepúlveda

Carrera: Ingeniería de Ejecución en Computación e Informática.

Concepción, mayo 2017

A modo de introducción podemos decir que el cliente es el que más envía mensajes por pantalla ya que el servidor solo envía un mensaje al final de responder una consulta. Destacar que los mensajes que envía el servidor son precedidos por la etiqueta “servidor>”, en el caso contrario los mensajes serían una instrucción *System.out.println()* desde el cliente

Preámbulo:

* Ingrese 1 si es usuario nuevo - Ingrese 2 si es usuario existente

El usuario ingresa la opción y esta es enviada al servidor. Si el usuario ingresa otros caracteres distintos de 1 o 2, se le pedirá que ingrese una opción válida.

* Si es un usuario nuevo:

Luego el cliente solicita al usuario un nombre y una clave.

* Ingrese su nombre de usuario – Ingrese su contraseña

El usuario ingresa su nombre y su clave y estos datos son enviados al servidor. El servidor valida si el nombre está disponible o no, en el caso de que el nombre ya exista se le solicita el nombre y clave nuevamente. Si el nombre no existe se imprimirá por pantalla:

* servidor> Bienvenido <nombre usuario>
* Si es un usuario existente:

Luego el cliente solicita al usuario un nombre y una clave.

* Ingrese su nombre de usuario – Ingrese su contraseña

El usuario ingresa su nombre y su clave y estos datos son enviados al servidor. El servidor valida si hay al menos un usuario registrado, si es que hay un error en la clave o si es que no se encuentra el usuario, así como si es que no hay usuarios registrados, si se da uno de esos casos se cerrará el cliente. Si la clave es correcta y el usuario existe se imprimirá por pantalla:

* servidor> Bienvenido <nombre usuario>

Consulta:

* servidor> ///Ingrese su opcion///
* servidor> 1. Consultar la hora y dia actual.
* servidor> 2. Consultar la hora y dia de la ultima consulta.
* servidor> 3. Listar los seudonimos de los usuarios registrados.
* servidor> 4. Enviar un mensaje a un usuario.
* servidor> 5. Consultar si hay algun mensaje a su nombre.
* servidor> 6. Borrar registro (implica borrar seudonimo y mensajes.
* servidor> 7. Finalizar sesion de consulta.
* ///Ingrese su opcion///

El servidor envía la lista de servicios que pueden ser solicitados al cliente y se le solicita al usuario la opción para ser enviada al servidor. La validación de que la opción sea del 1 al 7 se hace por el lado del cliente, si se ingresa una opción invalida se vuelve a solicitar.

* Al consultar la hora y dia actual

El servidor responde:

* servidor> La fecha es: <dia>-<mes>-<año> y son las <hora>:<minuto>:<segundo>
* Al consultar la hora y dia actual:

El servidor responde:

* servidor> La fecha es: <dia>-<mes>-<año> y son las <hora>:<minuto>:<segundo>
* Al consultar la hora y día de la última consulta:

El servidor responde:

* servidor> La ultima consulta fue el <dia>-<mes>-<año> a las <hora>:<minuto>:<segundo>
* Al listar los seudónimos de los usuarios registrados:

El servidor responde:

* servidor> Los usuarios registrados son:
* servidor> usuario1
* …
* servidor> usuarioN
* Al enviar mensaje a un usuario:

El servidor responde:

* servidor> Escriba destinatario

luego:

* servidor> Escriba mensaje

En el caso de que se escriba un usuario inexistente el servidor responderá:

* servidor> Usuario no encontrado

En el caso de que se escriba un mensaje superior a 30 caracteres el cliente hace esta validación y le responderá:

* Ingrese menos de 30 caracteres

Si el usuario existe y el número de caracteres es menor o igual a 30 servidor responderá:

* servidor> Mensaje enviado correctamente a <nombre usuario>
* Al consultar si hay algún mensaje:

El servidor responde:

* servidor> Tiene N mensaje(s).
* servidor> Ingrese cuantos mensajes desea mostrar.

En el caso de que se ingrese un número menor o igual a 0 o mayor al número de mensajes, se solicitará que ingrese nuevamente el numero ya que el servidor responderá:

* servidor> Ingrese opcion valida.

En el caso contrario se le preguntara al usuario:

* servidor> Ingrese opcion de cuales mensajes desea mostrar.
* servidor> 1. El/Los n primero(s) (más antiguo(s)).
* servidor> 2. El/Los n ultimo(s) (más nuevo(s)).

Las opciones son explícitas, ingresar 1 para mostrar los primeros mensajes y 2 para mostrar los últimos mensajes. Esta validación de caracteres está hecha en el lado del servidor.

Si el número es igual a la cantidad de mensajes estos se mostrarán sin ningún filtro, al momento de mostrar los mensajes se aplica el siguiente formato que aplica tanto para los mensajes filtrados como no filtrados:

* servidor> Mensaje> Remitente: <usuario> Contenido: <contenido>

Finalmente se pregunta al usuario:

* servidor> Desea eliminar los mensajes que leyo (s/n)?

Igualmente, como en el caso del filtrado las opciones son explícitas, ingresar S para eliminar los mensajes y N para eliminar mensajes. Esta validación de caracteres está hecha en el lado del servidor.

* Al borrar registro (seudónimo y mensajes):

El servidor responde:

* servidor> Ha sido eliminado correctamente.

Esto finaliza inmediatamente la sesión.

* Al finalizar sesión de consulta:
* servidor> Sesion finalizada

Término:

Al elegir una de las opciones mencionadas anteriormente el programa termina ahí, pero le da la siguiente opción al usuario:

* servidor> Desea realizar otra consulta? (s/n)

En este caso también se validan las respuestas por el lado del servidor.

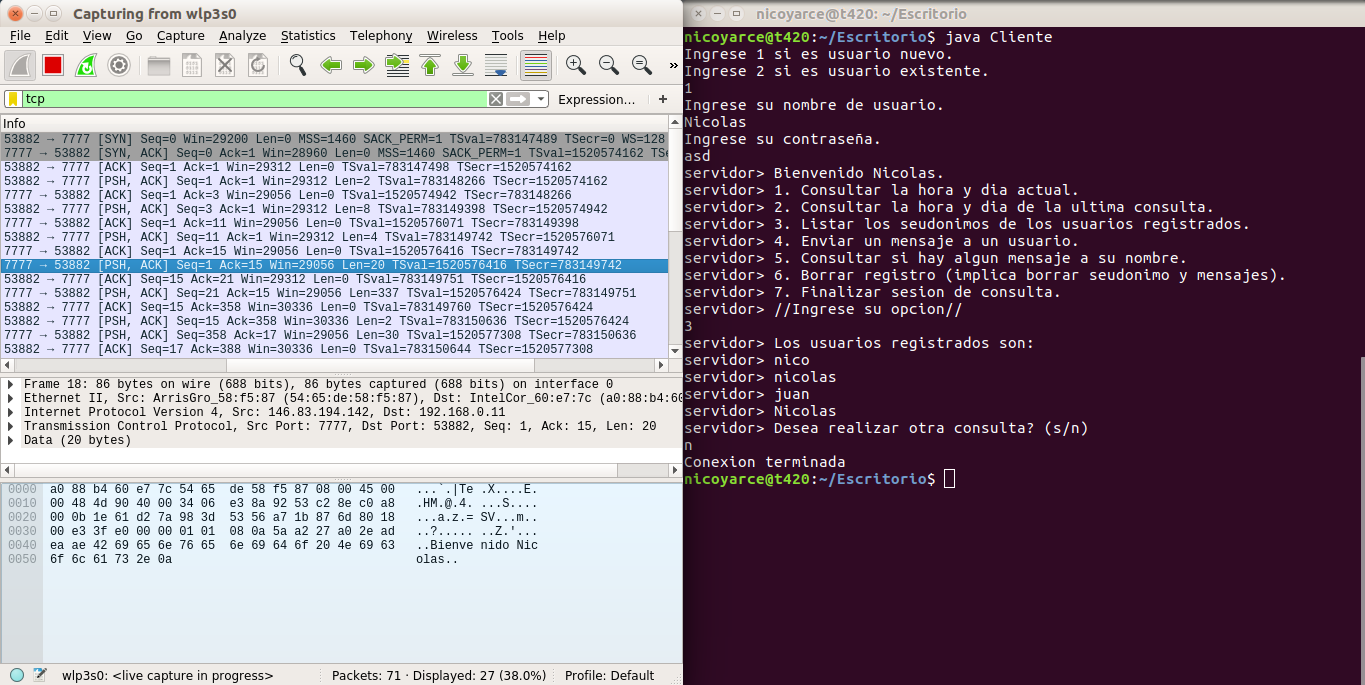
* Al responder s:

Se inicia nuevamente la sección de las consultas

* Al responder n el cliente imprime:
  + - Conexion terminada

Respecto a la utilización de protocolo TCP en el programa, esta decisión fue por tema de la necesidad que nace de que los mensajes no lleguen alterados ni que exista pérdida de paquetes, es decir se utiliza un protocolo confiable que resuelve el problema de integridad.

Capturas de pantalla de Wireshark:

Se muestran capturas del paquete y la ejecución del programa para corroborar el contenido.

