PRÁCTICA TELEMÁTICA

Práctica de Programación, Análisis, Diseño e Implementación de un chat multiusuario y temático utilizando Sockets

24/05/2013

UNIVERSIDAD EAFIT

NUBIA INÉS USUGA ESTEFANIA RODRÍGUEZ OSPINA

**TABLA DE CONTENIDO**

[**1.** **Diseño de protocolo:** 2](#_Toc357153207)

[Descripción 3](#_Toc357153208)

[Características 3](#_Toc357153209)

[Suposiciones del entorno 4](#_Toc357153210)

[Vocabulario de mensajes 4](#_Toc357153211)

[Codificación de mensajes 4](#_Toc357153212)

[Reglas de protocolo 4](#_Toc357153213)

[Implementación 5](#_Toc357153214)

[Código Fuente 5](#_Toc357153215)

[**2.** **Diseño del software** 6](#_Toc357153216)

[Diagrama de clases 6](#_Toc357153217)

[Diagrama de secuencias 7](#_Toc357153218)

[Diagrama de despliegue o componentes 8](#_Toc357153219)

[**3.** **Instalación** 8](#_Toc357153220)

[**4.** **Requerimientos de software.** 8](#_Toc357153221)

[**5.** **Proceso de ejecución** 9](#_Toc357153222)

[BIBLIOGRAFÍA 10](#_Toc357153223)

1. **Diseño de protocolo:**

Para el diseño del protocolo del chat multiusuario y temático, se debe seguir los siguientes pasos:

1. Definición del servicio
2. Suposiciones del entorno
3. Vocabulario de mensajes
4. Codificación de mensajes
5. Reglas de protocolo
6. Definición del Servicio

## Descripción

Realizar el análisis, diseño e implementación de un chat multiusuario y temático. Un usuario una vez se logea en el sistema (escribe un nombre), si el usuario no está logeado, lo dejará entrar, si hay alguien que ya ingreso con ese nombre le informa a través de un mensaje y le adiciona un 1 al nombre), entrará a un espacio en donde podrá realizar las siguientes operaciones:

* listar los usuarios activos y en que sala se encuentran (listar usuarios)
* Ver lista de las salas del sistema (lista salas)
* Entrar a una sala (Conectar)
* Salir del sistema (Salir)

Una vez entra a la sala de su preferencia, puede enviar mensajes de texto a la sala, mensaje que será enviado a cada usuario de la sala desde el servidor a cada cliente (opción 1, modo push). En síntesis puede realizar:

* Envió de mensajes (Enviar)
* Recibir (A través de la opción Push, donde lo emite el servidor).
* Salir del sistema (Salir)

## Características

Orientado a la conexión, punto a punto.

Confiable.

Con sesión.

Con seguridad (autenticación y verificación de la sala perteneciente)

## Suposiciones del entorno

El protocolo utilizará un protocolo confiable, concretamente sobre el protocolo TCP a través de la API de Sockets, esto implica que es confiable (detecta y corrige errores), hace control de flujo y control de errores.

## Vocabulario de mensajes

Se tienen 2 tipos de mensajes:

– cliente\_request” desde el cliente al servidor.

– “servidor\_response” desde el servidor al cliente. {cliente\_request, servidor\_response}

## Codificación de mensajes

• Se va a realizar una conversación entre dos clientes

• Se adoptará la siguiente convención:

– cliente\_request = “hola”

• “hola” es el mensaje a enviar, se acepta todo tipo de entrada por teclado

• Ejs: “cómo estas”, “qué haces”

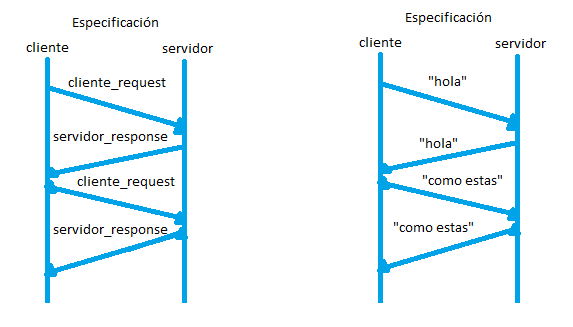
– servidor\_response = “hola”

• Se lleva el mensaje del cliente a todos los que están en la misma sala

• Ejs: “cómo estas”, “qué haces”

## Reglas de protocolo

Se utilizará un diagrama de tiempo para especificar el protocolo:



## Implementación

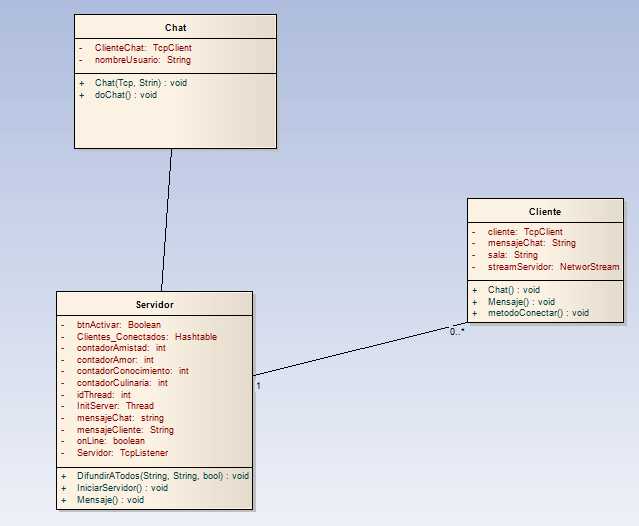
Se realiza la implementación del servicio del chat multiusuario y temático y el protocolo utilizando la herramienta Enterprese Architect para la parte del diseño y el lenguaje de programación c# para el desarrollo.

## Código Fuente

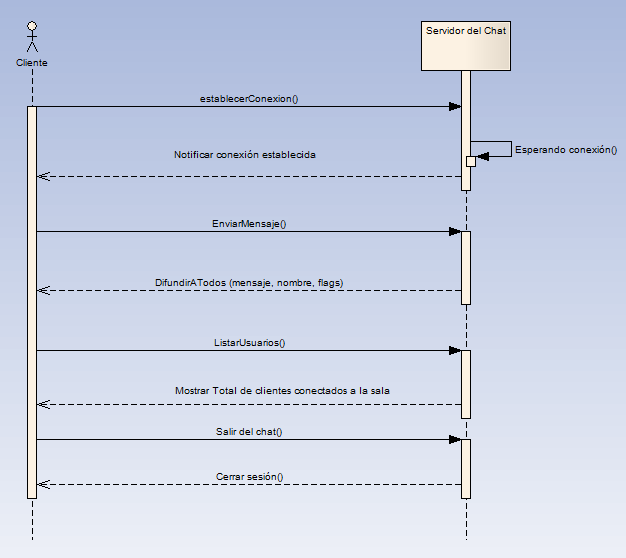
El código fuente se encuentra en el repositorio de Github: <https://github.com/nubiausuga/Chat>

1. **Diseño del software**

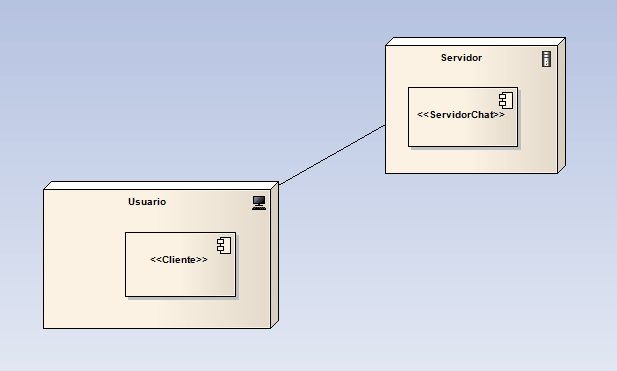
## Diagrama de clases



## Diagrama de secuencias



## Diagrama de despliegue o componentes



1. **Instalación**

Para la puesta en marcha de la aplicación solo es necesario tener el archivo .exe del actor que se desea ejecutar ya sea cliente o servidor.

1. **Requerimientos de software.**

Framework 4.0 de las librerías de .NET

Sistema operativo Windows XP o superior.

Conexión a internet, misma área local para todos los usuarios que quieran acceder al chat.

1. **Proceso de ejecución**

Para la ejecución de la aplicación se deben seguir los siguientes pasos:

1. Abrir el .exe llamado Servidor ubicado en la carpeta chat de la aplicación.
2. En la sección Dirección IP Indicar la dirección IP que se va a utilizar para la conexión.
3. En la sección Puerto Indicar la dirección del puerto que se va a utilizar para la conexión de todos los clientes con el servidor.
4. Dar clic en el botón de iniciar para iniciar el servidor al cual se van a conectar los clientes
5. Abrir el .exe llamado Cliente ubicado en la carpeta chat de la aplicación.
6. En la sección IP Servidor Indicar la dirección IP que utiliza el servidor para la conexión.
7. En la sección Puerto Indicar el número del puerto con el cual se va a establecer la conexión con el servidor.
8. En la sección Nombre Indicar el nombre con el cual se va a iniciar sesión en la sala, este sirve como autenticación de acceso.
9. En la sección Sala Indicar la sala a la que se desea conectar.
10. Dar clic en Conectar para establecer la conexión con el servidor.
11. Para enviar un mensaje a los demás clientes se inserta dicho mensaje en el cuadro de mensajes y se selecciona el botón enviar.
12. Para crear un nuevo cliente realizar nuevamente los pasos 5 a 10.
13. Para listar los usuarios que están en la sala dar clic al botón Listar Usuarios.
14. Para salir de la sala en la que está, clic en el botón salir, si se desea ingresar nuevamente a una sala realizar los pasos 5 al 9.
15. Si se desea parar la ejecución del servidor se debe dar clic al botón Cerrar en el servidor.

Ver detalles en el archivo adjunto: ProcesoEjecucion.doc

# BIBLIOGRAFÍA

Colaboradores de youtube. Marzo de 2012. Java - Sockets - Coding a Client-Server CHAT Room [en línea]. [Fecha de consulta: 17 mayo 2013]. Disponible en: <http://www.youtube.com/watch?v=Uo5DY546rKY>

Edwin Montoya. Mayo de 2012. Programación en Java Sockets. [Fecha de consulta: 17 mayo 2013].

Jorge Rubira Santos. Febrero de 2012. Chat con Java Video tutorial de Solo Programadores [en línea]. [Fecha de consulta: 18 mayo 2013]. Disponible en: <http://www.youtube.com/watch?v=-eYEMqL_6b4>

Java, Linux y programación. Marzo de 2008. Conexiones Cliente-Servidor mediante sockets en Java [en línea]. [Fecha de consulta: 18 mayo 2013]. Disponible en: <http://casidiablo.net/java-socket-chat-basico/>

Héctor Espinosa Blog. Mayo de 2011. Programar un Chat con C# (C# Chat) [en línea]. [Fecha de consulta: 19 mayo 2013]. Disponible en: [http://hectorea.com/blog/programar-un-chat-con-c-c-chat/#](http://hectorea.com/blog/programar-un-chat-con-c-c-chat/)