

# Chat en Java

Valdez Sampértiga José      Vargas Guanilo Javier

26 de marzo de 2008

## 1. Introducción

La comunicación se extiende a pasos agigantados y con ella los medios de comunicación, uno de ellos es el internet; y dentro de éste los software de mensajería instantánea tienen un valor especial en la comunicación.

Los software de mensajería instantánea o chat(nombre común) permiten además de comunicarse poder disfrutar de otras actividades como compartir archivos entre otras actividades.

El proyecto que a continuación se presenta consta de una aplicación de mensajería instantánea desarrollada en el lenguaje Java. Esta aplicación consta de 2 partes, la primera que es un servidor y la segunda que es un cliente; en el cliente podemos escoger entre hablar con todos los usuarios a la vez o hablar en privado con el usuario que nosotros deseemos.

## 2. Diagrama de Clases

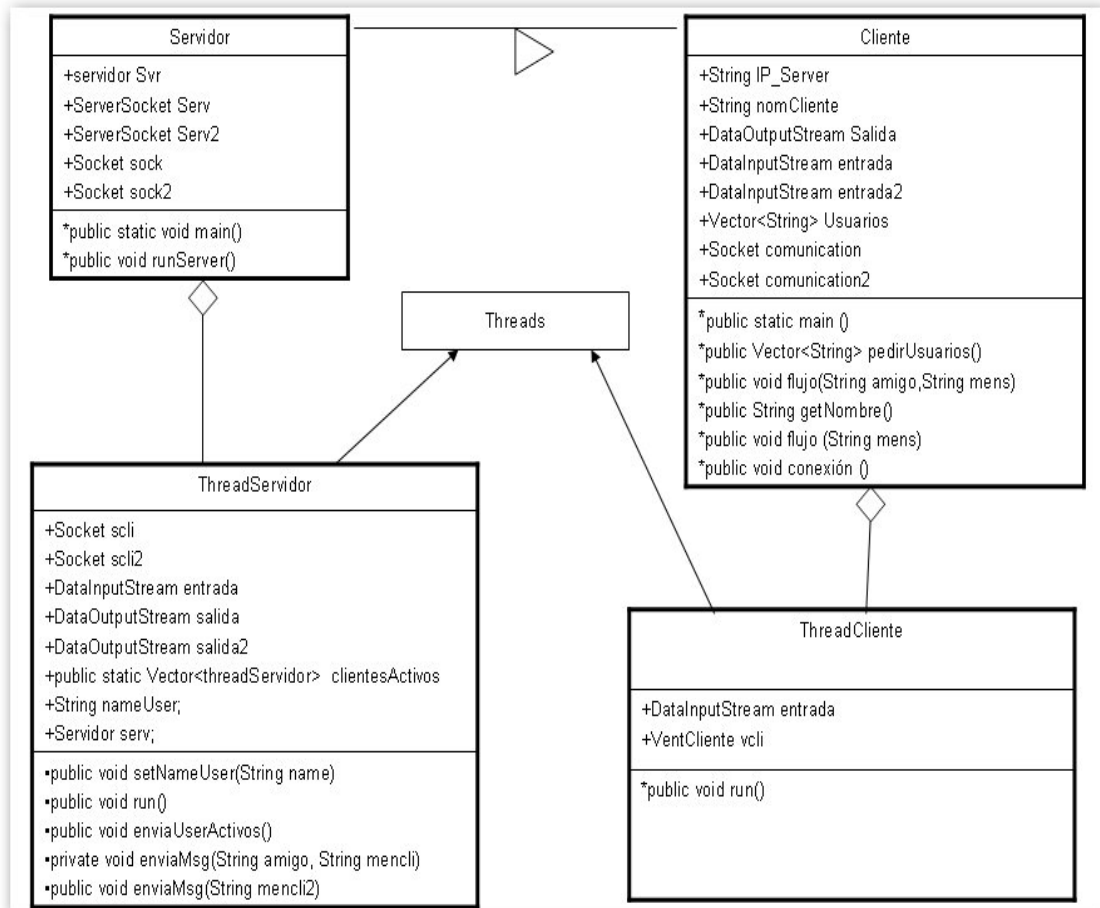


Figura 1: Diagrama de clases

### 3. Clases empleadas

#### 3.1. Clase:Servidor

##### 3.1.1. Descripción:

La clase servidor se mantiene a la escucha de cualquier petición de cliente que desee conectarse a él, si el cliente es aceptado el servidor establece 2 sockets de comunicación y los enlaza a un hilo entre el cliente y el servidor. Esta clase se hereda de la clase JFrame.

##### 3.1.2. Sumario de constructores

Constructor	Descripción
Servidor	Crea un frame para ver el estado del servidor

##### 3.1.3. Sumario de métodos

Nivel de acceso y tipo de retorno	Descripción
public static void main()	Función principal de la clase
public void runServer()	Función que mantiene en escucha al servidor

#### 3.2. Clase:Cliente

##### 3.2.1. Descripción:

La clase Cliente crea un flujo entre ella y el servidor para enviar y recibir mensajes, este flujo está afiliado a un hilo de comunicación.

##### 3.2.2. Sumario de constructores

Constructor	Descripción
Cliente	Declarado por default

### 3.2.3. Sumario de métodos

Nivel de acceso y tipo de retorno	Descripción
public static void main() public Vector<String> pedirUsuario() public void flujo() public void flujo(String mens) public String getNombre() public void conexion()	Funcion principal de la clase Pide los usuarios conectados al servidor carga un valor entero y el mensaje al servidor en este caso a un solo cliente Obtiene el nombre del cliente Crea la conexion con el servidor

## 3.3. Clase:threadCliente

### 3.3.1. Descripción:

Esta clase se encarga de gestionar los mensajes entre el hilo usuario e hilo servidor, hereda de la clase Thread

### 3.3.2. Sumario de constructores

Constructor	Descripción
threadCliente	Inicializa los atributos

### 3.3.3. Sumario de métodos

Nivel de acceso y tipo de retorno	Descripción
public void run()	Toma una accion de acuerdo al envio de Cliente

### 3.4. Clase:threadServidor

#### 3.4.1. Descripción:

Esta clase se encarga de gestionar los mensajes entre el hilo usuario e hilo servidor, hereda de la clase Thread

#### 3.4.2. Sumario de constructores

Constructor	Descripción
threadServidor	Añade nuevos clientes e imprime mensajes

#### 3.4.3. Sumario de métodos

Nivel de acceso y tipo de retorno	Descripción
public void run()	Abre sockets y envía usuarios activos
public void setNameUser(String name)	asigna una variable a otra
public void enviaUserActivos()	envía usuarios activos
private void enviaMsg(String amigo, String mencli)	Envia mensaje a todos
public void enviaMsg(String mencli2)	Envia mensaje a un solo cliente