



### **Ejercicio de chat en java**

---

Jorge Alberto Garduza Palacios

LICENCIATURA EN INGENIERIA EN SISTEMAS COMPUTACIONALES Y DISEÑO DE  
SOFTWARE, Instituto Universitario de Yucatán

2303040781: REDES

Perla Judith Landero Heredia

19 de octubre del 2025

---

The screenshot shows an IDE with three tabs: Cliente.java, Cliente2.java, and Servidor.java. The Cliente.java tab is active, displaying the following code:

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package cliente;
6
7   import java.io.*;
8   import java.net.*;
9
10  public class Cliente {
11
12      /**
13       * @param args the command line arguments
14       */
15      public static void main(String[] args) {
16
17          final String HOST = "localhost";
18          final int PUERTO = 3306;
19
20          try (Socket socket = new Socket (HOST, PUERTO)){
21
22              BufferedReader teclado = new BufferedReader (new InputStreamReader(System.in));
23              PrintWriter salida = new PrintWriter(socket.getOutputStream(), true);
24              BufferedReader entrada = new BufferedReader (new InputStreamReader(socket.getInputStream()));
```

The Output window at the bottom shows the following text:

```
run:
Servidor iniciado en el puerto3306
Cliente conectado
|
```

The screenshot shows the IDE output window with three tabs: Cliente (run), Cliente (run) #2, and Cliente (run) #3. The Cliente (run) tab is active, displaying the following text:

```
run:
Servidor iniciado en el puerto3306
Cliente conectado
Cliente dice:sfd
BUILD SUCCESSFUL (total time: 4 minutes 3 seconds)
```

```
10
11
12 public static void main(String[] args) {
13     final String HOST = "localhost";
14     final int PUERTO = 3306;
15
16     try (Socket socket = new Socket(HOST, PUERTO)) {
17         //Flujos de entrada y salida
18         BufferedReader teclado = new BufferedReader(new InputStreamReader(System.in));
19         PrintWriter salida = new PrintWriter(socket.getOutputStream(), true);
20         BufferedReader entrada = new BufferedReader(new InputStreamReader(socket.getInputStream()));
21
22         System.out.print("Escribe un mensaje: ");
23         String mensaje = teclado.readLine();
24         salida.println(mensaje);
25
26         String respuesta = entrada.readLine();
27         System.out.println("Servidor dice: " + respuesta);
28
29     } catch (IOException e) {
30         e.printStackTrace();
31     } // TODO code application logic here
32 }
```

```
run:
Escribe un mensaje: sfd
Servidor dice: Menssaje recibido:sfd
BUILD SUCCESSFUL (total time: 4 minutes 0 seconds)
```