



“CLIENTE Y SERVIDOR ”

(JESUS NICOMEDES DE LA CRUZ SANCHEZ)

(ING. SISTEMAS COMPUTACIONALES Y DISEÑO EN SOFTWARE), Instituto
Universitario de Yucatán

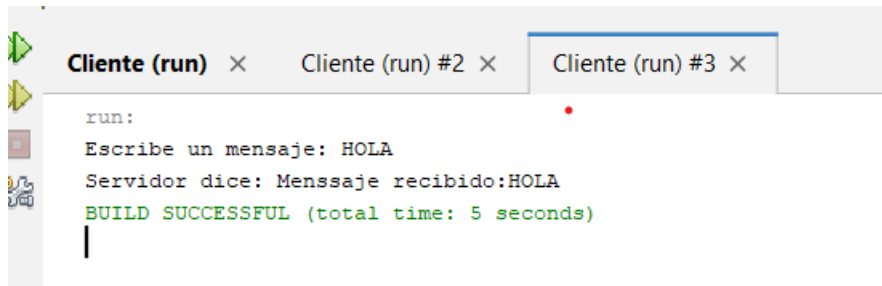
(2303040748): (REDES)

(PERLA ALEJANDRA LANDERO HEREDIA)

(19/10/2025)

```
Start Page x Cliente.java x Servidor.java x
Source History
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package cliente;
6  import java.io.*;
7  import java.net.*;
8
9  /**
10   *
11   * @author Jesus
12   */
13  public class Servidor {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          final int PUERTO = 3306;
21
22          try (ServerSocket servidor = new ServerSocket(port: PUERTO)) {
23              System.out.println("Servidor iniciado en el puerto" + PUERTO);
24              Socket socket = servidor.accept();
25              System.out.println(x: "Cliente conectado");
26
27              BufferedReader entrada = new BufferedReader(new InputStreamReader(in: socket.getInputStream()));
28              PrintWriter salida = new PrintWriter(out: socket.getOutputStream(), autoFlush: true);
29
30              String mensaje = entrada.readLine();
31              System.out.println("Cliente dice: " + mensaje);
32
33              salida.println("Mensaje recibido: " + mensaje);
34
35              socket.close();
36
37          } catch (IOException e) {
38              e.printStackTrace();
39          }
40      }
41  }
```

```
Start Page x Cliente.java x Servidor.java x
Source History
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package cliente;
6  import java.io.*;
7  import java.net.*;
8
9  /**
10   *
11   * @author Jesus
12   */
13  public class Cliente {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19
20          final String HOST = "localhost";
21          final int PUERTO = 3306;
22
23          try (Socket socket = new Socket(host: HOST, port: PUERTO)) {
24              //Flujos de entrada y salida
25              BufferedReader teclado = new BufferedReader(new InputStreamReader(in: System.in));
26              PrintWriter salida = new PrintWriter(out: socket.getOutputStream(), autoFlush: true);
27              BufferedReader entrada = new BufferedReader(new InputStreamReader(in: socket.getInputStream()));
28
29              System.out.print(s: "Escribe un mensaje: ");
30              String mensaje = teclado.readLine();
31              salida.println(x: mensaje);
32
33              String respuesta = entrada.readLine();
34              System.out.println("Servidor dice: " + respuesta);
35
36          } catch (IOException e) {
37              e.printStackTrace();
38          }
39      }
40  }
```



The screenshot shows an IDE terminal window with three tabs: "Cliente (run)", "Cliente (run) #2", and "Cliente (run) #3". The "Cliente (run)" tab is active. The terminal output is as follows:

```
run:
Escribe un mensaje: HOLA
Servidor dice: Menssaje recibido:HOLA
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
run:
Servidor iniciado en el puerto3306
Cliente conectado
Cliente dice:HOLA
BUILD SUCCESSFUL (total time: 10 seconds)
```