



“EJERCICIO DE CHAT”

MONTSERRAT OLAN LOPEZ

MODELO ENTIDAD-RELACIÓN, INSTITUTO UNIVERSITARIO DE YUCATÁN

2303040768: REDES II

MAESTRA ING. PERLA ALEJANDRA LANDERO HEREDIA

19 DE OCTUBRE DEL 2025

```
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  import java.io.*;
7  import java.net.*;
8
9  /**
10   *
11   * @author
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(PUERTO)) {
23              System.out.println("Servidor iniciado en el puerto" + PUERTO);
24              Socket socket = servidor.accept();
25              System.out.println("Cliente conectado");
26
27              BufferedReader entrada = new BufferedReader( new InputStreamReader (socket.getInputStream()));
28              PrintWriter salida = new PrintWriter(socket.getOutputStream(), true);
29
30              String mensaje = entrada.readLine();
31              System.out.println("Cliente dice:" + mensaje);
32
33              salida.println("Menssaje recibido:" + mensaje);
34
35              socket.close();
36
37          } catch (IOException e) {
```

Cliente (run)

```
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  import java.io.*;
7  import java.net.*;
8
9  /**
10   *
11   * @author
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, PUERTO)) {
24              //Flujos de entrada y salida
25              BufferedReader teclado = new BufferedReader(new InputStreamReader(System.in));
26              PrintWriter salida = new PrintWriter(socket.getOutputStream(), true);
27              BufferedReader entrada = new BufferedReader(new InputStreamReader(socket.getInputStream()));
28
29              System.out.print("Escribe un mensaje: ");
30              String mensaje = teclado.readLine();
31              salida.println(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  }
```

Servidor (run)

```
C:\Users\monts\OneDrive\Documentos\REDES 2\Cliente\src\cliente\Cliente.java

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, PUERTO)) {
24             //Flujos de entrada y salida
25             BufferedReader teclado = new BufferedReader(new InputStreamReader(System.in));
26             PrintWriter salida = new PrintWriter(socket.getOutputStream(), true);
27             BufferedReader entrada = new BufferedReader(new InputStreamReader(socket.getInputStream()));
28
29             System.out.print("Escribe un mensaje: ");
30             String mensaje = teclado.readLine();
31             salida.println(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
41
42
43
44     }
45
46
47
48     // TODO code application logic here
```

Output

```
run:
Escribe un mensaje: HOLA
Servidor dice: SERVIDOR: Bienvenido! Tu ID es #7
BUILD SUCCESSFUL (total time: 3 seconds)
```