



Instituto Oficial de Formación Profesional

Desarrollo de un Servidor de Chat Concurrente

Miguel Ceballos Sánchez
Félix Caballero Peña



Índice

Desarrollo de un Servidor de Chat Concurrente	1
1. Respuesta a la pregunta de la Fase 1.	2
2. Fase 3: El Servidor Multihilo (Concurrente)	3
3. Fase_4 Qué pasa si cierras la ventana del cliente sin escribir "FIN"?	5



1. Respuesta a la pregunta de la Fase 1.

Pregunta para el informe: ¿Qué ocurre con el Cliente 2? ¿Por qué no puede ni siquiera enviar su mensaje? Pista: Pensad en el "Hilo Principal" (Main Thread). ¿Si está ocupado durmiendo (sleep), puede volver a la línea accept() para atender al siguiente?

- El cliente 2 espera durante 15 segundos, que sería el tiempo estipulado que tarda el servidor en reconocer la conexión del cliente (uso del método sleep), para posteriormente expulsar al primer cliente y ejecutar el siguiente accept(), ya que el servidor es monohilo y no admite varios clientes al mismo tiempo.

2. Fase 3: El Servidor Multihilo (Concurrente)

The screenshot shows the IntelliJ IDEA interface. The project structure on the left includes a 'src' folder containing 'Fase_3' which contains 'Cliente', 'Multihilo', and 'ServidorPrincipal'. The 'Run' tool bar at the bottom has 'ServidorPrincipal' selected. The terminal window at the bottom shows the command 'C:\Users\Usuario1\.ideks\openjdk-25\bin\java.exe -javaagent:C:\Users\Usuario1\AppData\Local\Programs\IntelliJ IDEA Community Edition 2025.2.1\lib\idea_rt.jar -Dfile.encoding=UTF-8 -jar C:\Users\Usuario1\IdeaProjects\ChatConcurrente\out\production\ChatConcurrente\ChatConcurrente.jar' and the output 'Servidor concurrente iniciado...'. The code editor on the right shows the 'ServidorPrincipal.java' file:

```
1 package Fase_3;
2
3 import java.io.DataInputStream;
4 import java.io.DataOutputStream;
5 import java.io.IOException;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8
9 public class ServidorPrincipal {
10     static void main() {
11         ServerSocket server = null;
12         Socket socket = null;
13         DataInputStream in = null;
14         DataOutputStream out = null;
15
16         final int PUERTO = 5000;
17     }
18 }
```

Servidor iniciado esperando clientes



```
public class Cliente { public static void main() { try{ Socket socket = new Socket(HOST, PUERTO); in = new DataInputStream(socket.getInputStream()); out = new DataOutputStream(socket.getOutputStream()); Scanner s = new Scanner(System.in); System.out.println("Introduce tu nombre: "); String nombre = s.nextLine(); out.writeUTF(nombre); boolean salir = false; while(!salir){ System.out.println(nombre + ": "); } } catch (Exception e){ e.printStackTrace(); } } }
```

Cliente 1 conectándose al servidor

```
public class Cliente { public static void main() { try{ Socket socket = new Socket(HOST, PUERTO); in = new DataInputStream(socket.getInputStream()); out = new DataOutputStream(socket.getOutputStream()); Scanner s = new Scanner(System.in); System.out.println("Introduce tu nombre: "); String nombre = s.nextLine(); out.writeUTF(nombre); boolean salir = false; while(!salir){ System.out.println(nombre + ": "); } } catch (Exception e){ e.printStackTrace(); } } }
```

Cliente 2 conectándose al servidor



Project: ServidorChat [ChatConcurrente] C:\Users\Usuario1\IdeaProjects\ChatConcurrente

Code Editor: Cliente.java

```
public class Cliente { & mcs056
    static void main() { & mcs056
        try {
            Socket socket = new Socket(HOST, PUERTO);
            in = new DataInputStream(socket.getInputStream());
            out = new DataOutputStream(socket.getOutputStream());
            Scanner s = new Scanner(System.in);
            System.out.println("Introduce tu nombre: ");
            String nombre = s.nextLine();
            out.writeUTF(nombre);
            boolean salir = false;
            while(!salir){
                System.out.println(nombre + ": ");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Run: ServidorPrincipal

Output:

```
C:\Users\Usuario1\.jdks\openjdk-25\bin\java.exe -javaagent:C:\Users\Usuario1\AppData\Local\Programs\IntelliJ IDEA Community Edition 2025.2.1\lib\idea_rt.jar -Dfile.encoding=UTF-8
Servidor concurrente iniciado...
Cliente conectado: Miguel
Cliente conectado: Félix
```

Servidor con clientes conectados e identificados

Miguel:
Hola Félix, ¿cómo estás?
Servidor > Hola Félix, ¿cómo estás?

Félix:
Bien, ¿y tú?
Servidor > Bien, ¿y tú?

Mensaje de cada cliente desde su consola particular

```
Servidor concurrente iniciado...
Cliente conectado: Miguel
Cliente conectado: Félix
Miguel: Hola Félix, ¿cómo estás?
Félix: Bien, ¿y tú?
```

Consola del chat con la conversación de ambos clientes



3. Fase_4 Qué pasa si cierras la ventana del cliente sin escribir "FIN"?

```
Servidor concurrente iniciado...
Cliente conectado: Miguel desde /127.0.0.1
El cliente se ha desconectado inesperadamente
```

Mensaje de cierre forzoso cuando el cliente sale sin escribir "FIN"