



Desarrollo de un Servidor de Chat Concurrente

**Miguel Ceballos Sánchez
Félix Caballero Peña**



Índice

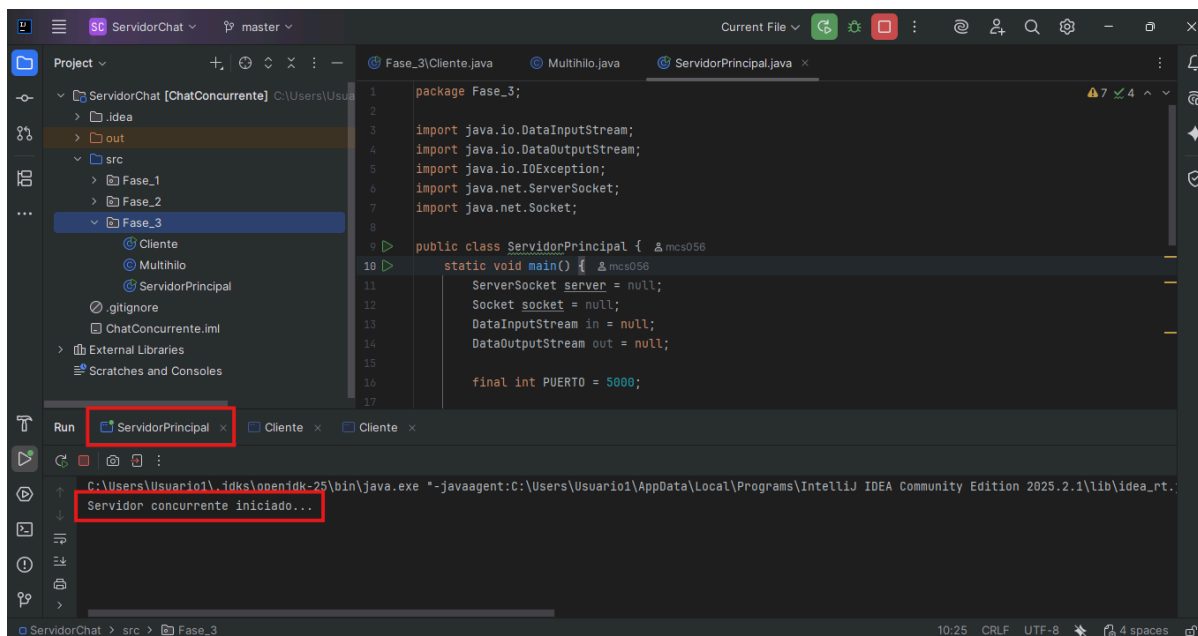
Desarrollo de un Servidor de Chat Concurrente	1
1. Respuesta a la pregunta de la Fase 1.	2
2. Fase 3: El Servidor Multihilo (Concurrente)	3
3. Fase_4 Qué pasa si cierras la ventana del cliente sin escribir "FIN"?	5

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)



```
package Fase_3;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class ServidorPrincipal { @mcs056
    static void main() { @mcs056
        ServerSocket server = null;
        Socket socket = null;
        DataInputStream in = null;
        DataOutputStream out = null;

        final int PUERTO = 5000;
    }
```

Run: ServidorPrincipal x Cliente x Cliente x

C:\Users\Usuario1\Idea\openjdk-25\bin\java.exe --javaagent:C:\Users\Usuario1\AppData\Local\Programs\IntelliJ IDEA Community Edition 2025.2.1\lib\idea_rt...

Servidor concurrente iniciado...

Servidor iniciado esperando clientes



```
public class Cliente {  
    static void main() {  
        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 + ":");  
        }  
    }  
}
```

Run: C:\Users\Usuario1\jdk-25\bin\java.exe "-javaagent:C:\Users\Usuario1\AppData\Local\Programs\IntelliJ IDEA Community Edition 2025.2.1\lib\idea_rt...

Introduce tu nombre:
Miguel
Miguel:

Cliente 1 conectándose al servidor

```
public class Cliente {  
    static void main() {  
        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 + ":");  
        }  
    }  
}
```

Run: C:\Users\Usuario1\jdk-25\bin\java.exe "-javaagent:C:\Users\Usuario1\AppData\Local\Programs\IntelliJ IDEA Community Edition 2025.2.1\lib\idea_rt...

Introduce tu nombre:
Félix
Félix:

Cliente 2 conectándose al servidor



```
public class Cliente {
    static void main() {
        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 + ": ");
        }
    }
}
```

Run console output:

```
Server 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

```
Server 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"