



INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo



UNIDAD DE APRENDIZAJE:

Desarrollo de Sistemas Distribuidos

**Tarea 1. Sistema distribuido que verifica si un
Número es primo**

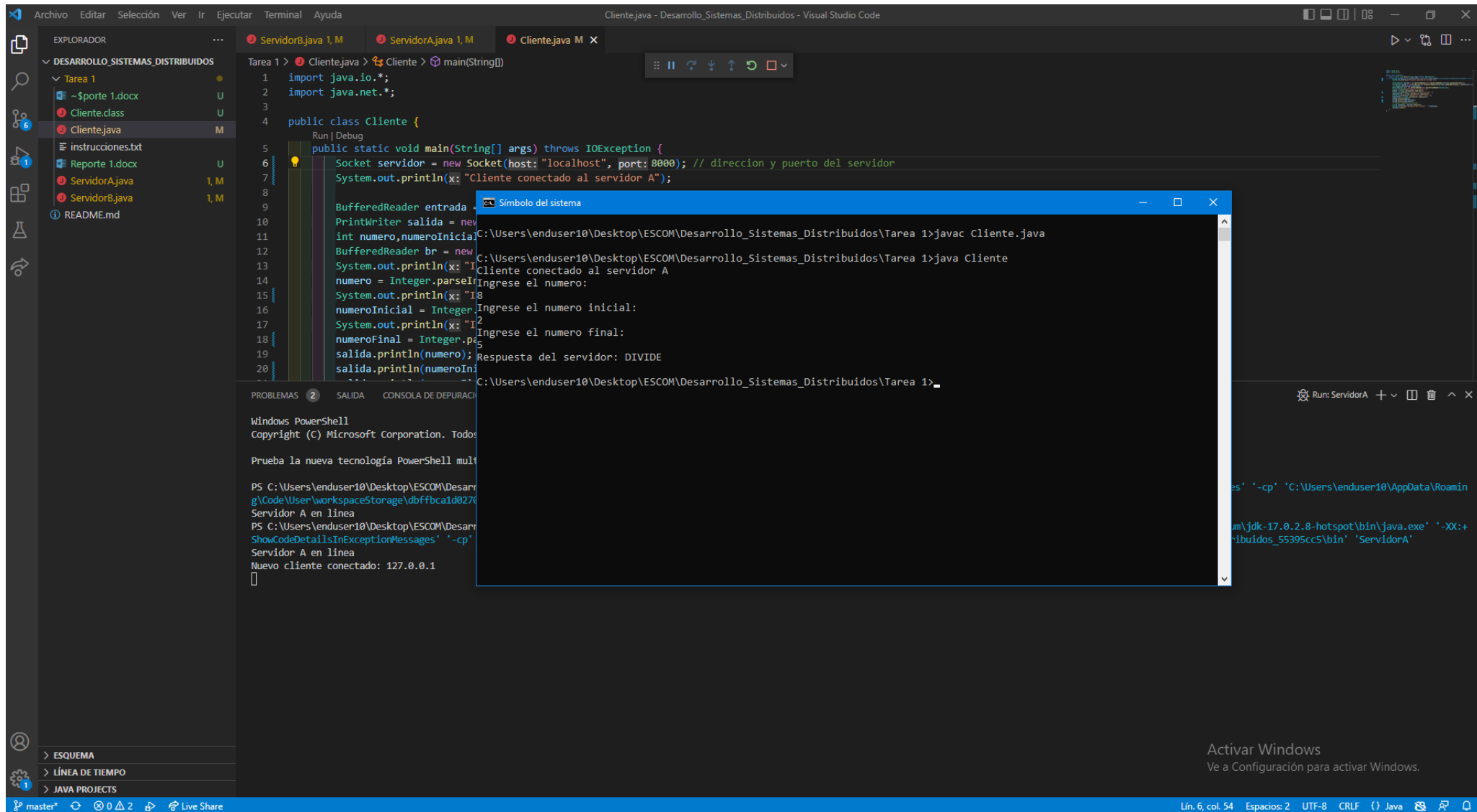
INTEGRANTES DEL EQUIPO

- Chávez Morones Ángel Uriel

NOMBRE DEL PROFESOR:

Carlos Pineda

Cliente enviando los números con respuesta del servidor



Visual Studio Code interface showing a Java project named "Desarrollo_Sistemas_Distribuidos". The Explorer pane on the left shows the project structure, including files like "Tarea 1", "Cliente.class", "Cliente.java", "Reporte 1.docx", "ServidorA.java", "ServidorB.java", and "README.md". The main editor displays the code for "Cliente.java", which includes imports for java.io.* and java.net.*, and a public class Cliente with a main method. The code is currently running, as indicated by the "Run | Debug" button in the top right. A "Símbolo del sistema" (System Symbol) window is open, showing the command prompt output for the Java application. The output shows the client connecting to the server, entering a number, and receiving a response. The bottom status bar indicates the current line and column (Lín. 6, col. 54) and the encoding (UTF-8).

```
1 import java.io.*;
2 import java.net.*;
3
4 public class Cliente {
5     public static void main(String[] args) throws IOException {
6         Socket servidor = new Socket(host: "localhost", port: 8080); // direccion y puerto del servidor
7         System.out.println(x: "Cliente conectado al servidor A");
8
9         BufferedReader entrada = new BufferedReader(new InputStreamReader(servidor.getInputStream()));
10        PrintWriter salida = new PrintWriter(servidor.getOutputStream());
11        int numero, numeroInicial;
12        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
13        System.out.println(x: "Cliente conectado al servidor A");
14        numero = Integer.parseInt(br.readLine());
15        System.out.println(x: "Ingrese el numero:");
16        numeroInicial = Integer.parseInt(br.readLine());
17        System.out.println(x: "Ingrese el numero inicial:");
18        numeroFinal = Integer.parseInt(br.readLine());
19        salida.println(numero);
20        salida.println(numeroInicial);
```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.
Prueba la nueva tecnología PowerShell multiplataforma.

PS C:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos> javac Cliente.java
PS C:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos> java Cliente
C:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos\Tarea 1> java Cliente
Cliente conectado al servidor A
Ingrese el numero:
5
Ingrese el numero inicial:
2
Ingrese el numero final:
4
Respuesta del servidor: NO DIVIDE
C:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos\Tarea 1>

Activar Windows
Ve a Configuración para activar Windows.

Ejecución del Servidor A

The screenshot displays the Visual Studio Code interface with the following components:

- EXPLORADOR (Left):** Shows the project structure for 'Desarrollo_Sistemas_Distribuidos'. The file 'Cliente.java' is selected under 'Tarea 1'.
- EDITOR (Center):** Displays the code for 'Cliente.java'. The code is a Java class named 'Cliente' with a 'main' method. It uses 'Socket' to connect to 'localhost' on port 8080 and 'BufferedReader' to read input from the user. The code is as follows:

```
1 import java.io.*;
2 import java.net.*;
3
4 public class Cliente {
5     public static void main(String[] args) throws IOException {
6         Socket servidor = new Socket(host: "localhost", port: 8080); // direccion y puerto del servidor
7         System.out.println(x: "Cliente conectado al servidor A");
8
9         BufferedReader entrada = new BufferedReader(new InputStreamReader(servidor.getInputStream()));
10        PrintWriter salida = new PrintWriter(new OutputStreamWriter(servidor.getOutputStream()), autoFlush: true);
11        int numero, numeroInicial, numeroFinal;
12        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
13        System.out.println(x: "Ingrese el numero: ");
14        numero = Integer.parseInt(br.readLine());
15        System.out.println(x: "Ingrese el numero inicial: ");
16        numeroInicial = Integer.parseInt(br.readLine());
17        System.out.println(x: "Ingrese el numero final: ");
18        numeroFinal = Integer.parseInt(br.readLine());
19        salida.println(numero);
20        salida.println(numeroInicial);
```
- TERMINAL (Bottom):** Shows the output of the command to run the server. The command is:

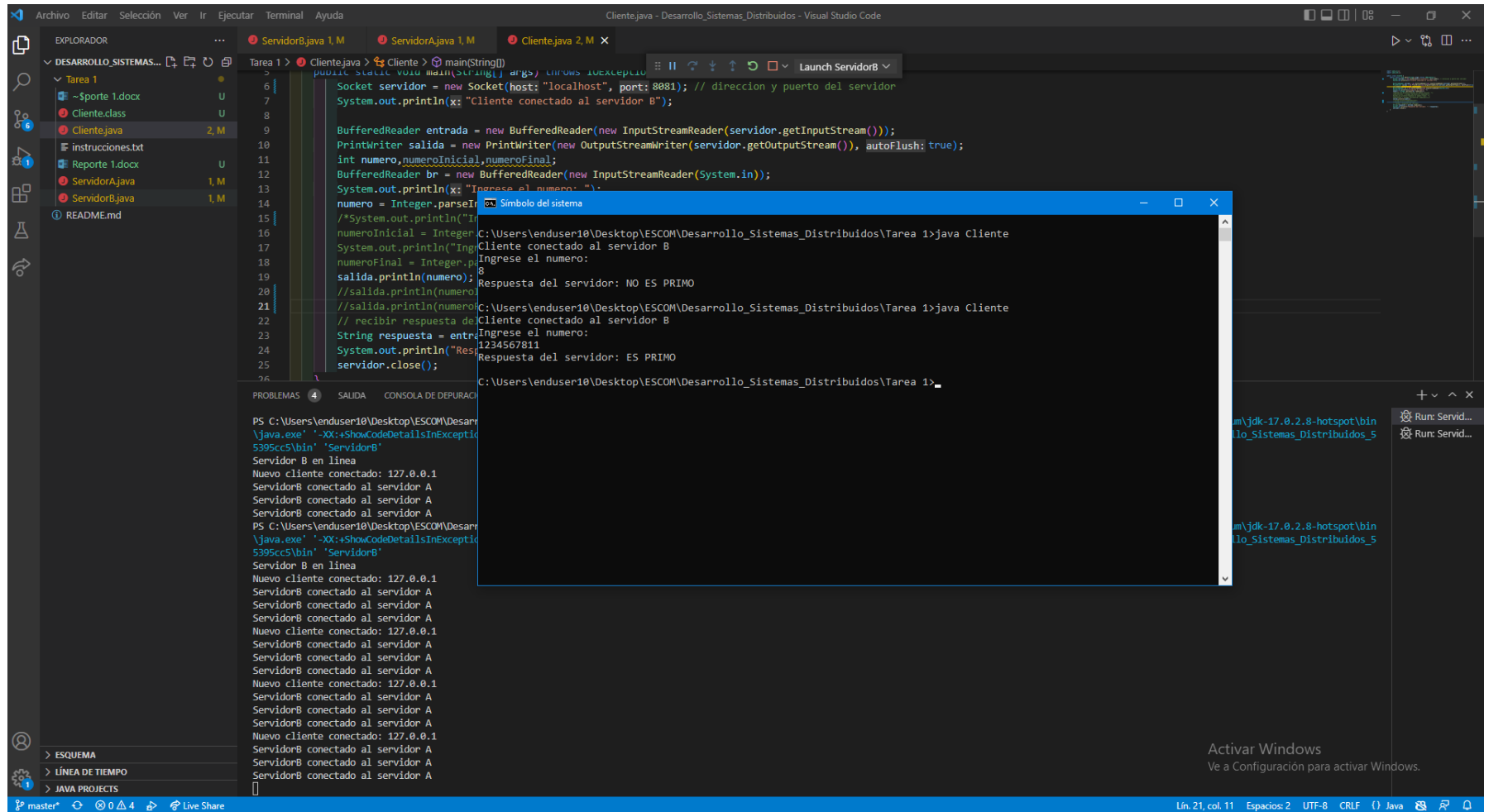
```
PS C:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\enduser10\AppData\Roaming\Code\User\workspaceStorage\dbffbcad1d8270c35e0894c19c9b75f6a\redhat.java\jdt_ws\Desarrollo_Sistemas_Distribuidos_55395cc5\bin' 'Servidora'
```

The output shows the server starting and receiving two client connections:

```
Servidor A en línea
Nuevo cliente conectado: 127.0.0.1
Nuevo cliente conectado: 127.0.0.1
```

Parte 2

Cliente



Servidor B

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

Cliente.java - Desarrollo_Sistemas_Distribuidos - Visual Studio Code

EXPLORADOR

- DESARROLLO_SISTEMAS_DISTRIBUIDOS
 - Tarea 1
 - ~\$porte 1.docx
 - Cliente.class
 - Cliente.java
 - Instrucciones.txt
 - Reporte 1.docx
 - ServidorA.java
 - ServidorB.java
 - README.md

Tarea 1 > Cliente > Cliente.java

```
public static void main(String[] args) throws IOException {
    Socket servidor = new Socket(host: "localhost", port: 8081); // direccion y puerto del servidor
    System.out.println(x: "Cliente conectado al servidor B");

    BufferedReader entrada = new BufferedReader(new InputStreamReader(servidor.getInputStream()));
    PrintWriter salida = new PrintWriter(new OutputStreamWriter(servidor.getOutputStream(), true));
    int numero, numeroInicial, numeroFinal;
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    System.out.println(x: "Ingrese el numero: ");
    numero = Integer.parseInt(br.readLine());
    /*System.out.println("Ingrese el numero inicial: ");
    numeroInicial = Integer.parseInt(br.readLine());
    System.out.println("Ingrese el numero final: ");
    numeroFinal = Integer.parseInt(br.readLine());*/
    salida.println(numero);
    //salida.println(numeroInicial);
    //salida.println(numeroFinal);
    // recibir respuesta del servidor
    String respuesta = entrada.readLine();
    System.out.println("Respuesta del servidor: " + respuesta);
    servidor.close();
}
```

PROBLEMAS 4 SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

```
PS C:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos> c:; cd 'c:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos'; & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.2-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\enduser10\AppData\Roaming\Code\User\workspaceStorage\dbffba1d0278c35e0894c19c9b75f6a\redhat.java\jdt_ws\Desarrollo_Sistemas_Distribuidos_5395cc5\bin' 'ServidorB'
Servidor B en línea
Nuevo cliente conectado: 127.0.0.1
ServidorB conectado al servidor A
ServidorB conectado al servidor A
ServidorB conectado al servidor A
Nuevo cliente conectado: 127.0.0.1
ServidorB conectado al servidor A
ServidorB conectado al servidor A
ServidorB conectado al servidor A
Nuevo cliente conectado: 127.0.0.1
ServidorB conectado al servidor A
ServidorB conectado al servidor A
ServidorB conectado al servidor A
Nuevo cliente conectado: 127.0.0.1
ServidorB conectado al servidor A
ServidorB conectado al servidor A
ServidorB conectado al servidor A
Nuevo cliente conectado: 127.0.0.1
ServidorB conectado al servidor A
ServidorB conectado al servidor A
ServidorB conectado al servidor A
```

Activar Windows
Ve a Configuración para activar Windows.

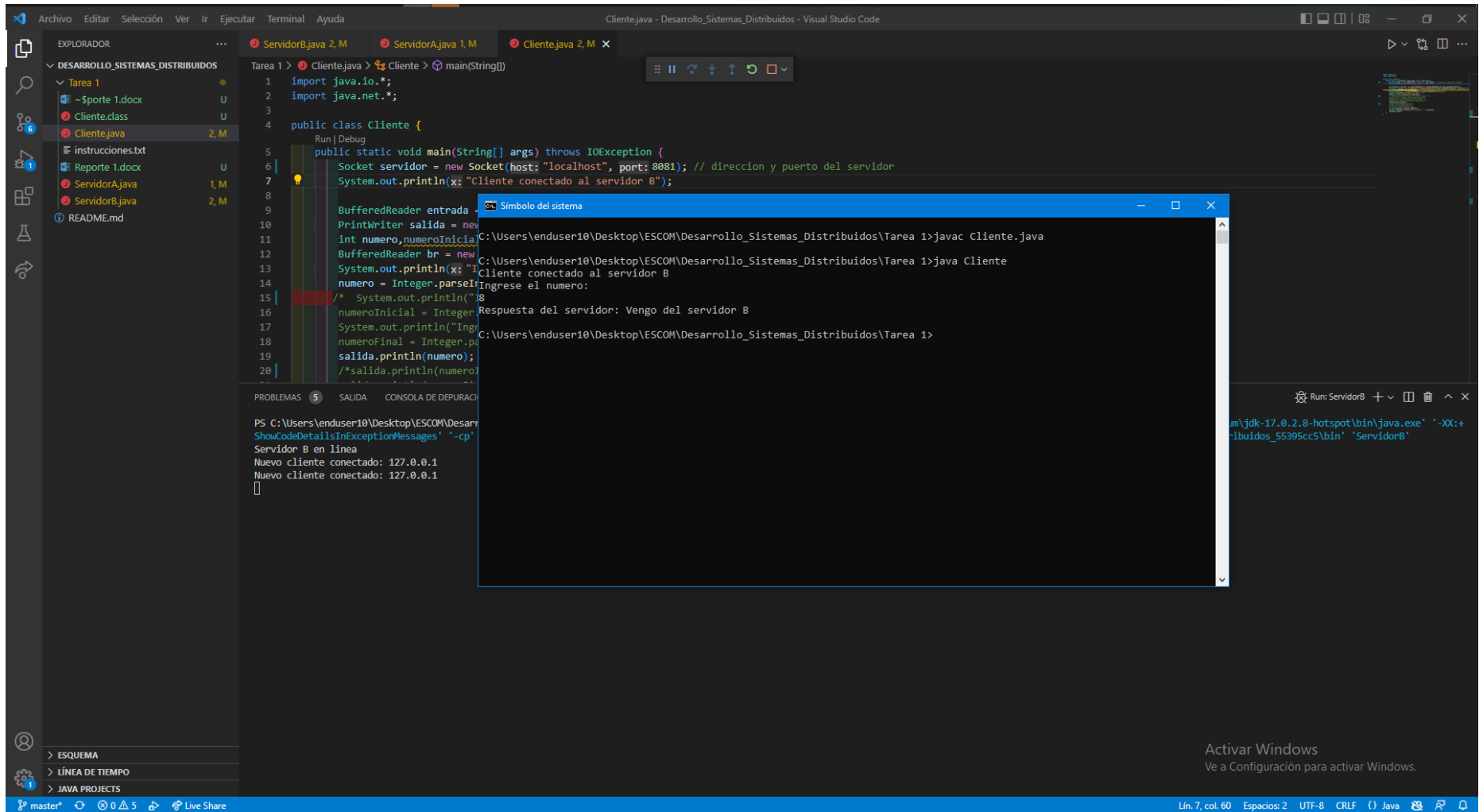
master 0 4 Live Share

Ln. 21, col. 11 Español UTF-8 CRLF Java

Servidor A

Parte 3

Cliente



Servidor

Visual Studio Code interface showing a Java project named "DESARROLLO_SISTEMAS_DISTRIBUIDOS". The Explorer sidebar on the left lists files: Tarea 1, ~\$porte 1.docx, Cliente.class, Cliente.java (2, M), Instrucciones.txt, Reporte 1.docx, ServidorA.java (1, M), ServidorB.java (2, M), and README.md. The main editor displays the code for "Cliente.java" (2, M), which includes imports for java.io.* and java.net.*, and a public class Cliente with a main method. The code sets up a Socket server on localhost:8081, creates a BufferedReader for input, and a PrintWriter for output. It prompts the user to enter a number and prints the result. The Terminal at the bottom shows the execution of the server and client, with output indicating a successful connection.

```
1 import java.io.*;
2 import java.net.*;
3
4 public class Cliente {
5     public static void main(String[] args) throws IOException {
6         Socket servidor = new Socket(host: "localhost", port: 8081); // direccion y puerto del servidor
7         System.out.println(x; "Cliente conectado al servidor B");
8
9         BufferedReader entrada = new BufferedReader(new InputStreamReader(servidor.getInputStream()));
10        PrintWriter salida = new PrintWriter(new OutputStreamWriter(servidor.getOutputStream()), autoFlush; true);
11        int numero, numeroInicial, numeroFinal;
12        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
13        System.out.println(x; "Ingrese el numero: ");
14        numero = Integer.parseInt(br.readLine());
15        /* System.out.println("Ingrese el numero inicial: ");
16        numeroInicial = Integer.parseInt(br.readLine());
17        System.out.println("Ingrese el numero final: ");
18        numeroFinal = Integer.parseInt(br.readLine());*/
19        salida.println(numero);
20        /*salida.println(numeroInicial);
```

Terminal output:

```
PS C:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos> c:; cd "c:\Users\enduser10\Desktop\ESCOM\Desarrollo_Sistemas_Distribuidos"; & "C:\Program Files\Eclipse Adoptium\jdk-17.0.2-hotspot\bin\java.exe" "-XX:+ShowCodeDetailsInExceptionMessages" "-cp" "C:\Users\enduser10\AppData\Roaming\Code\User\workspaceStorage\d8ffbc1d0270c35e0894c19c9b75f6a\redhat.java\jdk_ws\Desarrollo_Sistemas_Distribuidos_55395cc5\bin" "ServidorB"
Servidor B en línea
Nuevo cliente conectado: 127.0.0.1
Nuevo cliente conectado: 127.0.0.1
```

Conclusiones

Con esta tarea pudimos ver el funcionamiento correcto de los servidores multithread a la vez que repasamos de la programación de dichos servidores. Aunque la forma de encontrar si es primo es fuerza bruta por lo que esto puede provocar un lento funcionamiento del mismo programa nos sirvió para recordar aspectos básicos de los servidores al igual que del código usado y como se puede implementar.