

### Actividad 3.5.2

Crea un programa servidor que pueda atender hasta 3 clientes. Debe enviar a cada cliente un mensaje indicando el número de cliente que es. Este número será 1, 2 o 3. El cliente mostrará el mensaje recibido. Cambia el programa para que lo haga para N clientes, siendo N un parámetro que tendrás que definir en el programa.

Clase Servidor:

```
public class Servidor {

    public static void main(String[] args) throws IOException {

        int numeroPuerto = 6000;// Puerto

        int contador = 1;

        ServerSocket servidor = new ServerSocket(numeroPuerto);

        Scanner sc = new Scanner(System.in);

        System.out.println("Indique cuantos clientes quiere aceptar: ");

        int numeroClientes = sc.nextInt();

        for (int i = 0; i < numeroClientes; i++) {

            Socket clienteConectado = servidor.accept();

            System.out.println("Mensaje enviado al cliente " + contador);

            // CREO FLUJO DE ENTRADA DEL CLIENTE

            InputStream entrada = null;

            entrada = clienteConectado.getInputStream();

            DataInputStream flujoEntrada = new DataInputStream(entrada);

            //Recibo del usuario el numero y lo elevo al cuadrado

            // CREO FLUJO DE SALIDA AL CLIENTE

            OutputStream salida = null;

            salida = clienteConectado.getOutputStream();

            DataOutputStream flujoSalida = new DataOutputStream(salida);

            // ENVIO UN MENSAJE AL CLIENTE

            flujoSalida.writeUTF("Conectado con cliente " + contador);

            contador++;
        }
    }
}
```

```
// CERRAR STREAMS Y SOCKETS
```

```
    entrada.close();
```

```
    flujoEntrada.close();
```

```
    salida.close();
```

```
    flujoSalida.close();
```

```
    clienteConectado.close();
```

```
}
```

```
servidor.close();
```

```
}// main
```

```
}// fin
```

Clase Cliente:

```
public class Cliente {
```

```
    public static void main(String[] args) throws IOException {
```

```
        String Host = "localhost";
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int numero;
```

```
        int Puerto = 6000;//puerto remoto
```

```
        System.out.println("PROGRAMA CLIENTE INICIADO....");
```

```
        Socket cliente = new Socket(Host, Puerto);
```

```
// CREO FLUJO DE SALIDA AL SERVIDOR :
```

```
    DataOutputStream flujoSalida = new DataOutputStream(cliente.getOutputStream());
```

```
// CREO FLUJO DE ENTRADA AL SERVIDOR
```

```
    DataInputStream flujoEntrada = new DataInputStream(cliente.getInputStream());
```

```
// EL SERVIDOR ME ENVIA UN MENSAJE
```

```
    System.out.println("Recibiendo del Servidor: \n\t"
```

```
        + flujoEntrada.readUTF());
```

```
// CERRAR STREAMS Y SOCKETS
```

```
    flujoEntrada.close();
```

```

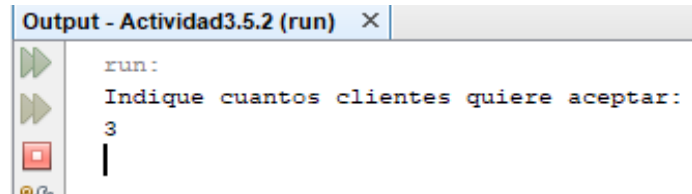
    flujoSalida.close();

    cliente.close();

} //main
}

```

- Ejecuto primero la clase Servidor:

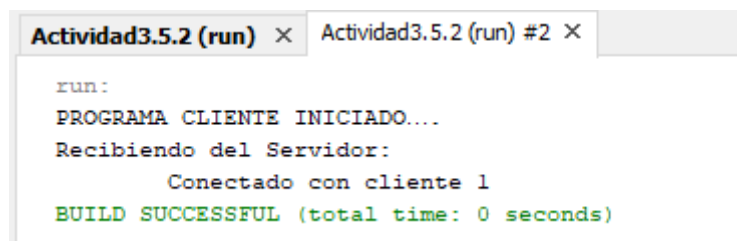


```

Output - Actividad3.5.2 (run) ×
run:
Indique cuantos clientes quiere aceptar:
3
|

```

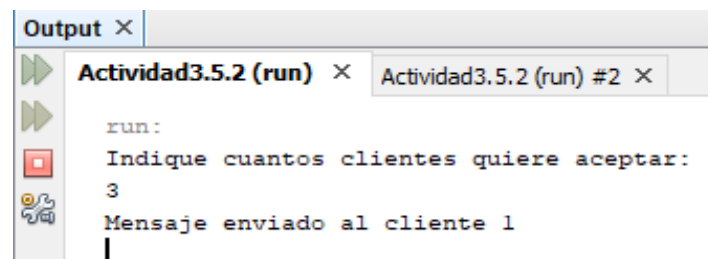
- Ahora ejecuto la clase Cliente y al volver al Servidor aparece lo siguiente:



```

Actividad3.5.2 (run) × Actividad3.5.2 (run) #2 ×
run:
PROGRAMA CLIENTE INICIADO....
Recibiendo del Servidor:
    Conectado con cliente 1
BUILD SUCCESSFUL (total time: 0 seconds)

```

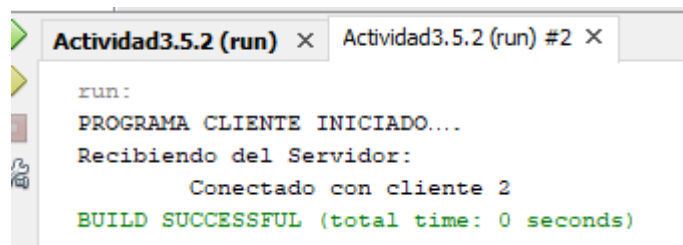


```

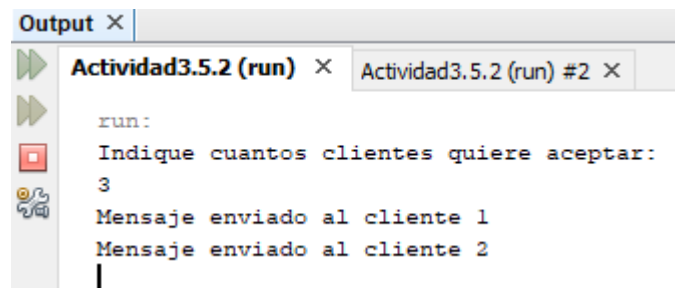
Output ×
Actividad3.5.2 (run) × Actividad3.5.2 (run) #2 ×
run:
Indique cuantos clientes quiere aceptar:
3
Mensaje enviado al cliente 1
|

```

- Y si ejecuto de nuevo el cliente:

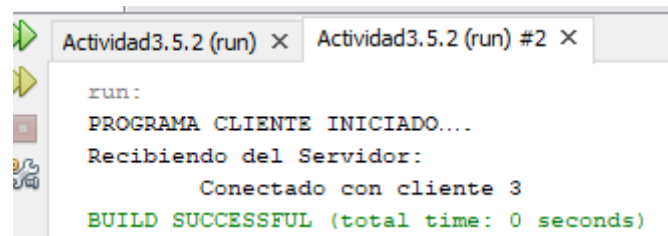


```
run:
PROGRAMA CLIENTE INICIADO....
Recibiendo del Servidor:
    Conectado con cliente 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

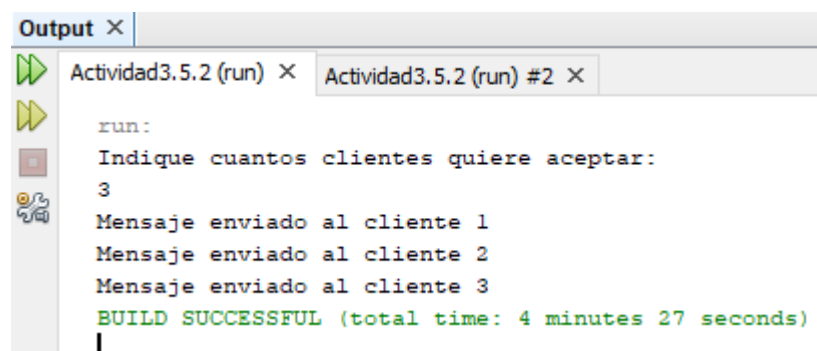


```
run:
Indique cuantos clientes quiere aceptar:
3
Mensaje enviado al cliente 1
Mensaje enviado al cliente 2
|
```

- Y así, hasta la tercera vez al menos en este caso ya que le puse que lo hiciera 3 veces:



```
run:
PROGRAMA CLIENTE INICIADO....
Recibiendo del Servidor:
    Conectado con cliente 3
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
run:
Indique cuantos clientes quiere aceptar:
3
Mensaje enviado al cliente 1
Mensaje enviado al cliente 2
Mensaje enviado al cliente 3
BUILD SUCCESSFUL (total time: 4 minutes 27 seconds)
|
```