

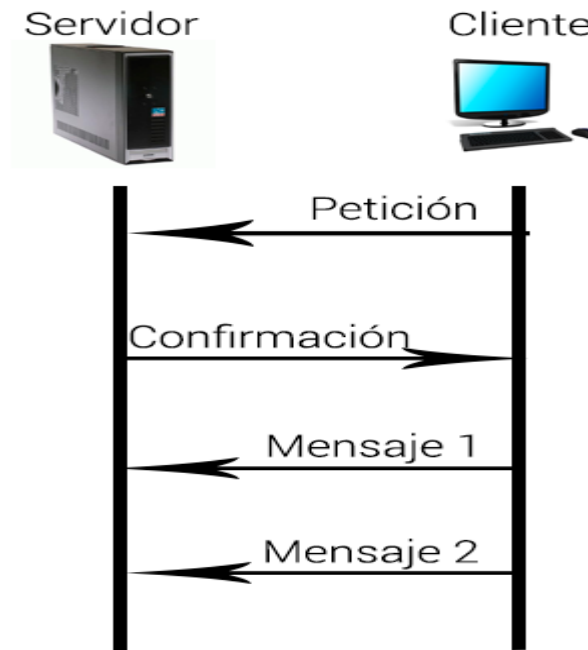
SISTEMAS DISTRIBUIDOS

Prof. José Alejandro
Reyes Ortiz

Servidores con Sockets

Servidor

- Espera las conexiones del cliente (en un lugar claramente definido).



Cliente

- Lanza las peticiones a la maquina donde se está ejecutando el servidor, y al lugar donde está esperando el servidor (el puerto(s) específico que atiende)



Sockets

- Comunicación entre dos computadoras de manera ordenada.
- Intercambio de mensajes de manera fiable utilizando el protocolo TCP.
- Un *socket* son una implementación independiente de la plataforma.

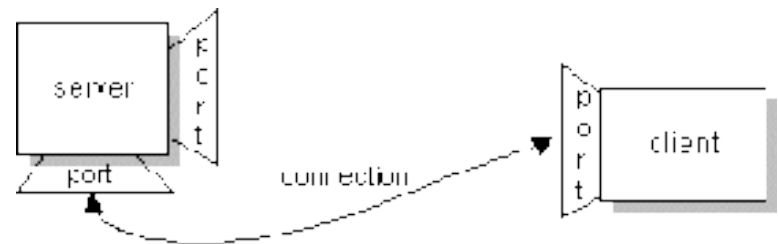
Sockets

- Comunicación entre dos computadoras de manera ordenada.
- Intercambio de mensajes de manera fiable utilizando el protocolo TCP.
- Un *socket* son una implementación independiente de la plataforma.

Envío de datos con Sockets

Servidor

- El servidor creará un socket, utilizando `ServerSocket`, le asignará un puerto y una dirección.
- Realiza el `accept()` para quedar a la espera de llamadas.

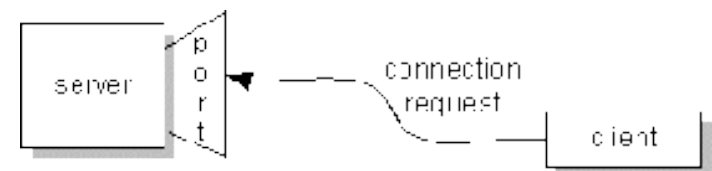


Envío de datos con Sockets

Ciente

- Un cliente desea establecer una conexión, creará un socket y establecerá una conexión al puerto establecido.

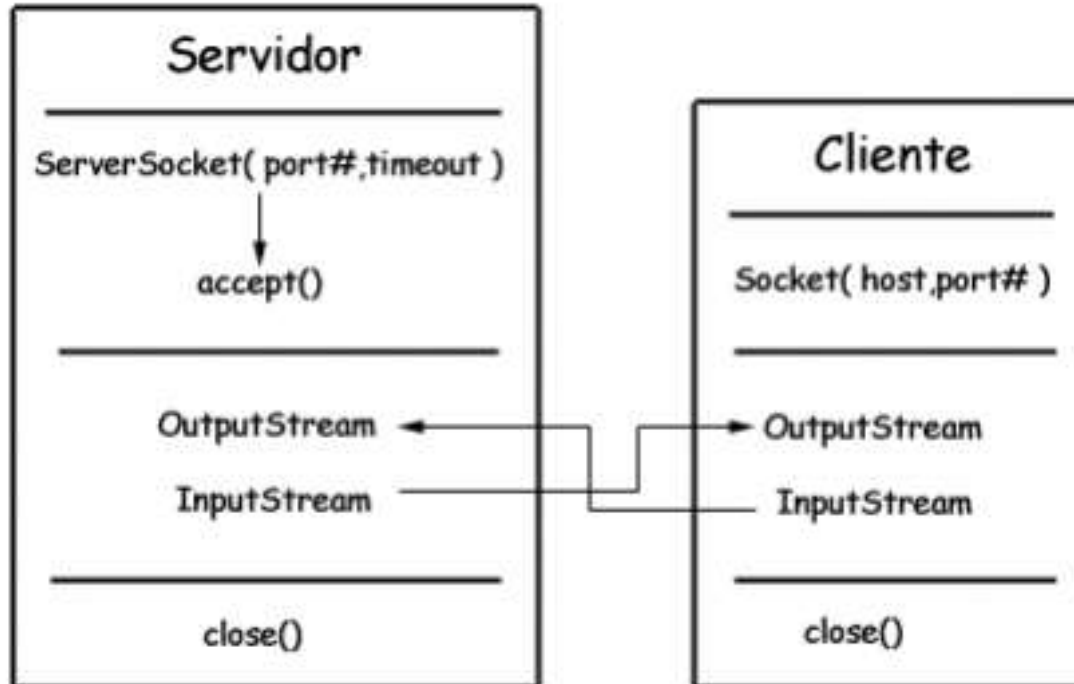
Se crea una conexión real y se mantendrá hasta su liberación mediante `close()`.



Sockets en Java

- A través de las clases del paquete `java.net`, los programas Java pueden utilizar TCP para comunicarse a través de una red.

Envío de datos con Sockets



Clase ServerSocket

Esta clase implementa un servidor sockets, el cual se encuentra en espera de peticiones entrantes sobre una red de computadoras. Esta clase desempeña algunas operaciones basada en el requerimiento, y entonces retorna un resultado el solicitante (cliente).

ServerSocket(int port)

Crear un servidor socket solo especificando el número de Puerto. 50 conexiones máximas.

ServerSocket(int port, int backlog)

Crear un servidor socket solo especificando el número de puerto y número de conexiones máximas.

ServerSocket(int port, int backlog, **InetAddress** bindAddr)

Crear un servidor con el puerto especificado, número de conexiones permitidas, y la dirección IP local a conectarse.

Clase ServerSocket



```
ServerSocket unServerSocket = new ServerSocket(4444);  
unServerSocket.accept();
```

Clase Socket

Esta clase implementa clientes sockets. Permite la comunicación del nodo cliente hacia otra máquina.

`Socket(String host, int port)`

Crear un socket de canal de comunicación mediante la especificación de un número de Puerto y del nombre de la maquina destino.

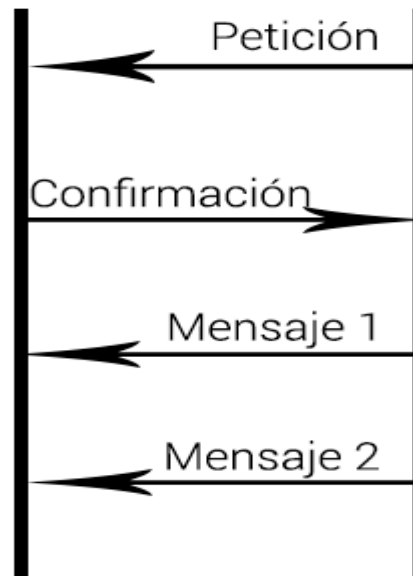
```
Socket unSocket = new Socket("localhost", 4444);
```

Ejemplo

Servidor



Cliente



Sockets en Java

Las clases *DataOutputStream* y *DataInputStream*

- Estas clases se utilizan para enviar y recibir datos a través del flujo (canal de comunicación) creado con *Socket* y *ServerSocket*.
- Estas clases son útiles para leer (read) y escribir (write) datos primitivos en estos canales.

Sockets en Java

Las clases *DataOutputStream* y *DataInputStream*

- Definen varios métodos para la lectura (read) de datos primitivos específicos:

```
boolean readBoolean();  
byte readByte();  
int readUnsignedByte();  
short readShort();  
int readUnsignedShort();  
char readChar();  
int readInt();  
String readLine();  
long readLong();  
float readFloat();  
double readDouble();
```

Sockets en Java

Las clases *DataOutputStream* y *DataInputStream*

- Definen varios métodos para escribir (write) datos primitivos específicos:

```
void writeBoolean(boolean v);  
void writeByte(int v);  
void writeBytes(String s);  
void writeShort(int v);  
void writeChars(String s);  
void writeChar(int v);  
void writeInt(int v);  
void writeLong(long v);  
void writeFloat(float v);  
void writeDouble(double v);
```