

Práctica 5

Redes y Sistemas Distribuidos
Grado de Ingeniería del Software (Grupo A)



UNIVERSIDAD
DE MÁLAGA





ServerSocket

Socket para un servidor orientado a la conexión

Constructores:

- `ServerSocket(int puerto)`: crea un socket pque recibe peticiones por el puerto indicado (0 = cualquier disponible). Cola de 50.
- `ServerSocket(int puerto, int count)`: igual que el previo pero con el tamaño de la cola (máximo número de peticiones en espera)
- `ServerSocket()`: crea un socket no vinculado (se debe usar `bind` con un puerto antes de usarse)

Métodos:

- `Socket accept()`: saca una petición de la cola de peticiones (si no hay se bloquea) y crea un socket conectado al cliente
- `bind(SocketAddress sa)`: vincula un socket a un puerto
- `close()`: libera el puerto y los recursos



Socket

Socket (conectado) para una comunicación TCP

Constructores:

- `Socket(String ip, int puerto)`: crea un socket TCP y lo conecta a la dirección IP y puertos indicados
- `Socket(InetAddress ip, int puerto)`: igual que el previo

Métodos:

- `InputStream getInputStream()`: devuelve el stream para la recepción de datos
- `OutputStream getOutputStream()`: devuelve el stream para el envío de datos
- `close()`: libera el puerto y los recursos



Envío y recepción a través de Socket

`getInputStream` y `getOutputStream` obtienen los flujos asociados

Recepción:

```
BufferedReader r = new BufferedReader(  
    new InputStreamReader(s.getInputStream())  
);
```

- `BufferedReader` dispone de los métodos `read` (recibe uno o varios bytes) y `readLine` (recibe líneas completas y devuelve un `String`)

Envío:

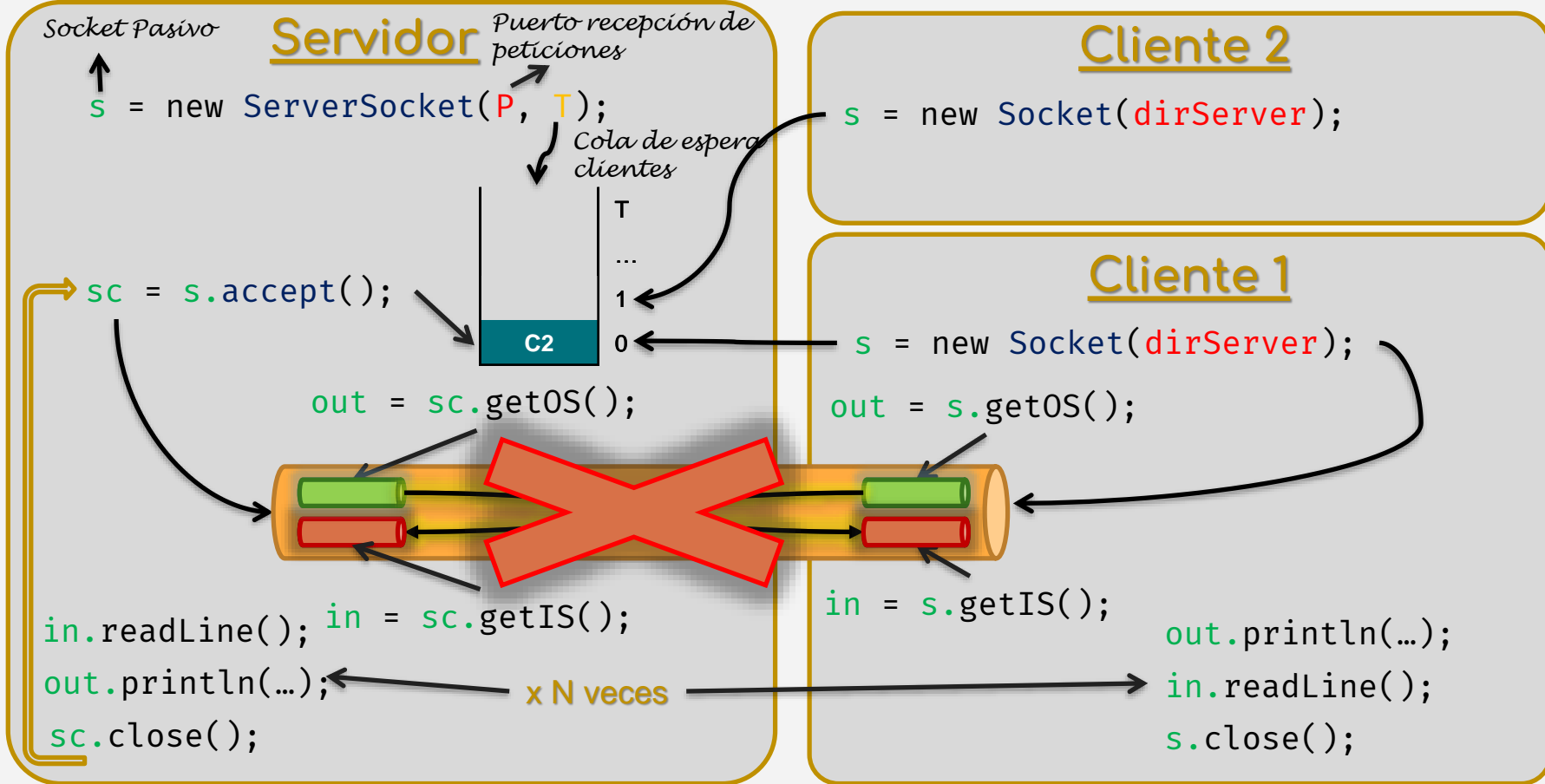
```
PrintWriter s = new PrintWriter(socket.getOutputStream(), true);
```

- `PrintWriter` dispone de los métodos `write` (envía un conjunto de líneas) y `println` / `print` (envía `String` completas con / sin `'\n'`)



Funcionamiento de C/S TCP

5



Protocolo de aplicación

Mismo servicio que con el protocolo UDP (extracción de textos)

Cliente:

- El usuario pedirá líneas al cliente
- Si lo leído es correcto (empieza por un dígito) lo enviará al servidor y esperará la respuesta (texto descifrado)
- Si el usuario escribe algo sin dígito inicial, enviará el texto FINISH y esperará el OK del servidor

Servidor:

- Recibirá múltiples líneas del cliente
- Cuando el cliente envíe un FINISH, el servidor responderá con OK y cerrará la conexión con el cliente
- Si el cliente se desconecta sin avisar, el servidor debe cerrar el socket de forma correcta y liberar los recursos
- Cola de espera de clientes tamaño 1