

Práctica 4

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



UNIVERSIDAD
DE MÁLAGA



Esquema de funcionamiento de C/S UDP

Cliente (ipC):

```
// Un puerto libre
s = CrearSocketUDP();
```

ipS , puertoS

```
...
s.enviar(mensaje, destino);
```

```
...
s.recibir(¿mensaje?, ¿origen ?);
```

```
...
// Liberar puerto / recursos
s.close();
```

Servidor (ipS):

```
// Un puerto concreto
s = CrearSocketUDP(puertoS);
```

```
...
s.recibir(¿mensaje?, ¿origen?);
```

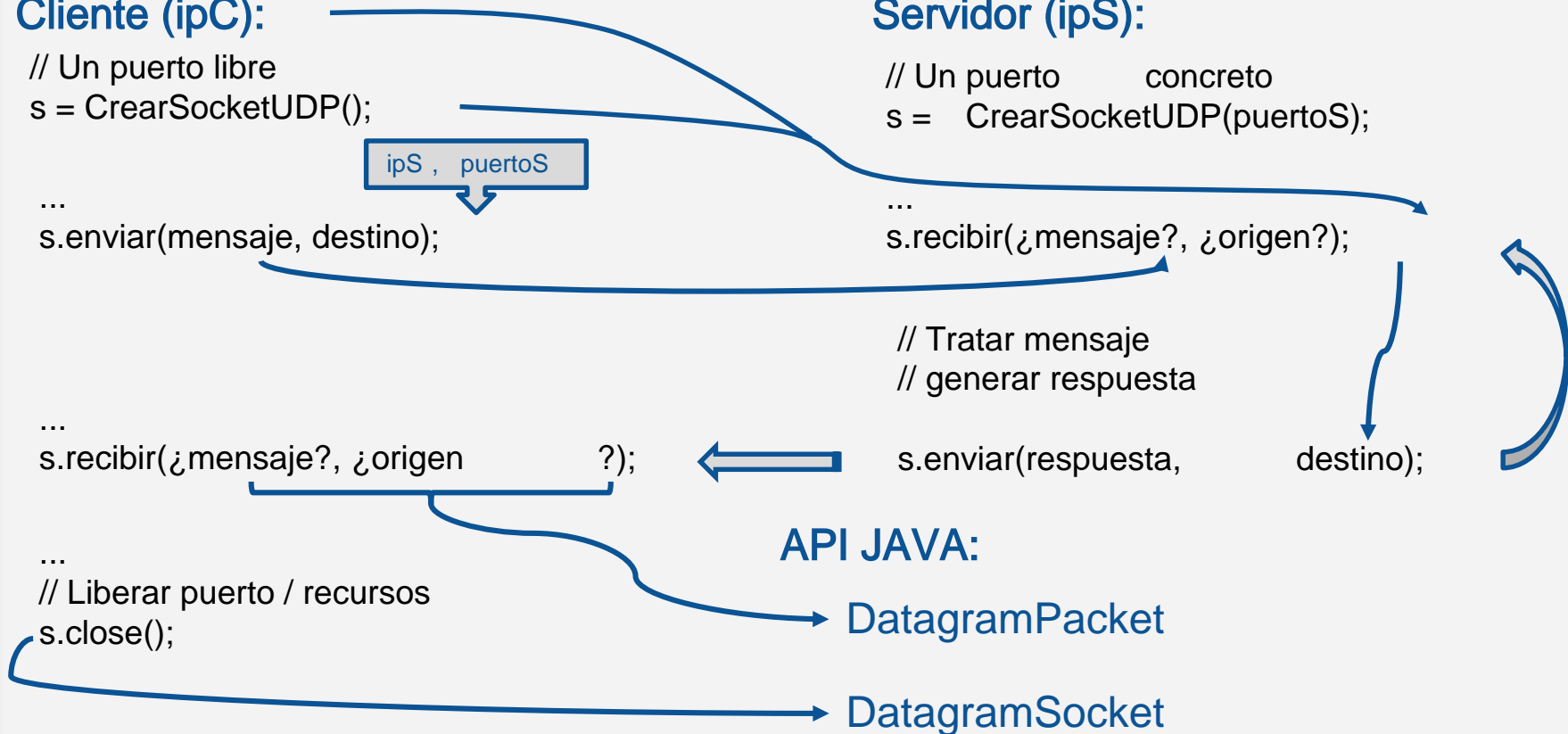
```
// Tratar mensaje
// generar respuesta
```

```
s.enviar(respuesta, destino);
```

API JAVA:

DatagramPacket

DatagramSocket





DatagramSocket

3

Socket para el envío y recepción de datagramas

Constructores:

- `DatagramSocket()`: crea un socket UDP con puerto efímero
- `DatagramSocket(int puerto)`: crea un socket UDP con un puerto concreto

Métodos:

- `send(DatagramPacket p)`: envía el datagrama
- `receive(DatagramPacket p)`: recibe el datagrama
- `close()`: libera el puerto y los recursos



DatagramPacket

4

Datagrama a enviar (mensaje, dirección) o memoria para recibir

Constructores:

- `DatagramPacket(byte [] b, int length)`: crea un datagrama para la recepción (zona de memoria donde se guarda el mensaje)
- `DatagramPacket(byte [] b, int length, InetAddress ip, int puerto)`: crea un datagrama para enviar

`InetAddress.getByName(nombre);`

Métodos:

- `InetAddress getAddress()`: IP al que se envía o de la que se recibe
- `int getPort()`: puerto al que se envía o del que se recibe
- `byte [] getData()`: datos a enviar o recibidos
- `int getLength()`: cantidad de datos enviados o recibidos



Protocolo de aplicación

Servidor:

- Recibe líneas de texto
- Las “capitaliza” y se las devuelve
- Nunca acaba

Cliente:

- Pide al usuario múltiples líneas de texto
- Cada una de ellas se las pasa al servidor y las recibe capitalizadas
- El usuario puede acabar este proceso escribiendo TERMINAR

Común:

- Los textos nunca superarán los 400 bytes
- Deben mostrar por la entrada estándar su estado
- Deben ser configurables por argumentos de la línea de comandos:
puerto (servidor) y dirección donde encontrar al servidor (cliente)