

**T3. Programació  
de  
comunicacions  
en xarxa. UDP.**

*Bloc 2*

PSP

# Continguts

- Sockets UDP

# Sockets UDP

Fins ara, hem utilitzat sockets TCP.

Recordem que els sockets TCP ens garantien el lliurament i l'ordre de paquets

En els casos en què no siga essencial ordre i lliurament podem utilitzar un altre tipus de protocol → El UDP.

En UDP, els paquets seran anomenats datagrames.

Els datagrames s'envien des d'un procés emissor a un altre receptor sense que s'haja establert prèviament connexió, sense justificant de recepció ni intents.

# Sockets UDP

Qualsevol procés que desitge enviar o rebre missatges ha de crear primer un connector (associat a IP i port)

El servidor enllaçarà el seu connector a un port conegut pels seus clients. El client enllaçarà el seu connector a qualsevol port local lliure.

Quan el receptor rep missatge s'obté: missatge + adreça IP de l'emissor + port de l'emissor, amb el que podrà enviar resposta corresponent.

Sockets UDP són més simples i eficients que els UDP però no està garantit el lliurament.

# Classes Sockets UDP

Java.net proporciona les classes :

- DatagramPacket
- DatagramSocket

Seran les que utilitzarem per implementar els sockets UDP

# Classe DatagramPacket. Constructor

La classe DatagramPacket conté constructors que podem utilitzar per crear datagrames. Un d'ells és:

```
DatagramPacket(byte[] buf, int length, InetAddress address, int port);
```

→ Constructor per l'enviament de datagrames. S'especifica la cadena de bytes a enviar (buf), la longitud (length), l'adreça (address) i el port (port).

# Classe DatagramPacket. Mètodes importants

<code>byte[] getData()</code>	Retorna les dades del buffer. És a dir el contingut del datagrama.
<code>int getLength()</code>	Retorna la longitud de les dades a enviar o rebre.
<code>void setData(byte[] buf)</code>	Estableix el buffer de dades per a aquest paquet
<code>void setLength(int length)</code>	Estableix la longitud d'aquest paquet

# Classe DatagramSocket

Dóna suport a enviament i recepció de paquets UDP

Mètodes importants són:

<code>void send(DatagramPacket p)</code>	Envia un datagrama des del socket
<code>void receive(DatagramPacket p)</code>	Rep un datagrama des del socket



# Exemple Servidor UDP bàsic 1/2

---

```
1 import java.net.*;
2 import java.io.*;
3
4 public class U3Exemple3_UDPServidor {
5     public static void main(String args[]) {
6
7         DatagramSocket dSocket = null;
8
9         if (args.length < 1) { // REBEM EN args EL PORT
10             System.out.println("Uso: java UDPServer <número de port>");
11             System.exit(1);
12         }
```

# Exemple Servidor UDP bàsic 2/2

```
14     try {
15         int socket_no = Integer.valueOf(args[0]).intValue(); // GUARDE EL NÚMERO DE PORT
16         dSocket = new DatagramSocket(socket_no);
17         byte[] missatgeRebut = new byte[1000];
18
19         while (true) {
20             DatagramPacket dpRebut = new DatagramPacket(missatgeRebut, missatgeRebut.length);
21             dSocket.receive(dpRebut);
22             System.out.println("Rep del client: " + new String(dpRebut.getData()));
23
24             // ENVIE EL DATAGRAMA a dpRebut.getAddress() I AL PORT dpRebut.getPort() (DEL QUE HE REBUT)
25             // ENVIE EL MATEIX MISSATGE QUE REP: dpRebut.getData()
26             DatagramPacket dpResposta = new DatagramPacket(dpRebut.getData(), dpRebut.getLength(),
27                 dpRebut.getAddress(), dpRebut.getPort());
28             dSocket.send(dpResposta);
29         }
30     } catch (SocketException e) {
31         System.out.println("Socket: " + e.getMessage());
32     } catch (IOException e) {
33         System.out.println("IO: " + e.getMessage());
34     } finally {
35         if (dSocket != null) // SI EL SOCKET EXISTEIX
36             dSocket.close(); // TANQUE
37     }
38 }
39 }
```

# Exemple Client UDP bàsic 1/2

```
1 import java.net.*;
2 import java.io.*;
3
4 public class U3Exemple3_UDPClient {
5     // REBEM EN args MISSATGE, HOST DESTINACIÓ I PORT
6     public static void main(String args[]) {
7
8         DatagramSocket dSocket = null;
9
10        // CONTROL ENTRADA ARGUMENTS
11        if (args.length < 3) {
12            System.out.println("Utilització: java U3Exemple3_UDPClient <missatge> <nom del Host> <número de port>");
13            System.exit(1);
14        }
```

# Exemple Client UDP bàsic 2/2

```
15 try {
16
17     // ENVIAMENT DEL DATAGRAMA
18     dSocket = new DatagramSocket();
19     byte[] missatgeEnviat = args[0].getBytes();
20     InetAddress aHost = InetAddress.getByName(args[1]); // RECUPERE EL HOST DES DE L'ARGUMENT
21     int serverPort = Integer.valueOf(args[2]).intValue(); // RECUPERE EL PORT DES DE L'ARGUMENT
22     DatagramPacket dpEnviament = new DatagramPacket(missatgeEnviat, args[0].length(), aHost, serverPort); // DATAGRAMA A ENVIAR
23     dSocket.send(dpEnviament); // ENVIE EL DATAGRAMA
24
25     // RECEPCIÓ DEL DATAGRAMA
26     byte[] missatgeRebut = new byte[1000];
27     DatagramPacket dpResposta = new DatagramPacket(missatgeRebut, missatgeRebut.length);
28     dSocket.receive(dpResposta); // REP EL DATAGRAMA
29     System.out.println("Resposta: " + new String(dpResposta.getData()));
30
31 } catch (SocketException e) {
32     System.out.println("Socket: " + e.getMessage());
33 } catch (IOException e) {
34     System.out.println("IO: " + e.getMessage());
35 } finally {
36     if (dSocket != null) // SI EL SOCKET EXISTEIX
37         dSocket.close(); // TANQUE
38 }
39 }
40 }
```

# Utilització Client-Servidor UDP bàsic

## Servidor

```
C:\U3Exemple3_UDPClietServidor\servidor>
C:\U3Exemple3_UDPClietServidor\servidor>javac U3Exemple3_UDPServidor.java

C:\U3Exemple3_UDPClietServidor\servidor>java U3Exemple3_UDPServidor
Uso: java U3Exemple3_UDPServidor <número de port>

C:\U3Exemple3_UDPClietServidor\servidor>java U3Exemple3_UDPServidor 5000
```

↓  
Servidor en espera.  
Quan rep serveix i  
mostra recepció

```
C:\U3Exemple3_UDPClietServidor\servidor>java U3Exemple3_UDPServidor 5000
Rep del client: Hola
```

## Client

```
C:\U3Exemple3_UDPClietServidor\client>javac U3Exemple3_UDPCliet.java

C:\U3Exemple3_UDPClietServidor\client>java U3Exemple3_UDPCliet
Utilització: java U3Exemple3_UDPCliet <missatge> <nom del Host> <número de port>

C:\U3Exemple3_UDPClietServidor\client>java U3Exemple3_UDPCliet Hola localhost 5000
Resposta: Hola
```

Client envia missatge i acaba.