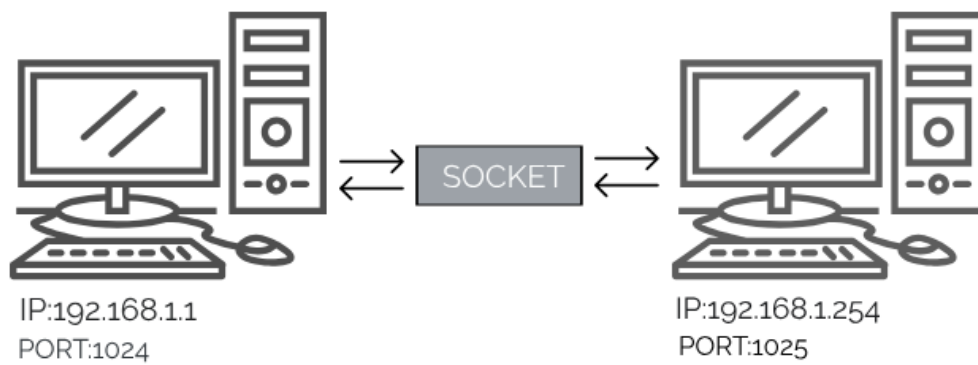


## Exercises UD03



1. **Training**
2. **Verification**
3. **Application activities**
4. **Extension**
5. **Information sources**

# 1. Training

---

## 1. Exchange of text messages between client and server through TCP sockets.

Create a client application and a server application that exchange text messages basing the communication on TCP sockets.

The output generated by the server execution is as follows:

```

1  (Server) Waiting for connections...
2  (Server) Connection established.
3  (Server) Opening text channels...
4  (Server) Open text channels.
5  (Server) Reading message...
6  (Server) Message read.
7  (Server) Received message: Message sent from the client
8  (Server) Sending message...
9  (Server) Message sent.
10 (Server) Closing text channels.
11 (Server) Text channels closed.
12 (Server) Closing connections...
13 (Server) Connections closed.
14
```

The output generated by the client execution is as follows:

```

1  (Client) Establishing connection...
2  (Client) Connection established.
3  (Client) Opening text channels...
4  (Client) Open text channels.
5  (Client) Sending message...
6  (Client) Message sent.
7  (Client) Message read.
8  (Client) Received message: Message sent from the server
9  (Client) Closing text channels.
10 (Client) Closed text channels.
11 (Client) Closing connections...
12 (Client) Connections closed.
13
```

## 2. Remote reading of files.

Develop a Java application that allows you to read a text file located on another computer through sockets. The steps of the process will be the following:

- Client program: asks the user for the name of a file including its full path within the server's file system.
- Client program: sends the name and path of the file to the server.
- Server program: reads the content of the file and sends it to the client.
- Client program: shows the content on the screen.

### 3. Data transfer by UDP

Develop a Java application that transmits numbers from a client to a server using UDP sockets. The process steps are as follows:

- Client program: through a loop it generates and sends 10,000 messages with the content "Message: message\_number" taking message\_number values between 0 and 9999.
  - Client program: when it has sent all the numbers it sends the string «END».
  - Server program: receives the messages and stores them in a file.
  - Server program: when it receives the string «END» execution ends.
  - Once the execution is finished, check if all the datagrams have arrived in the same order that they were sent. If the execution is carried out on the same computer, it is likely that all the messages will arrive in the correct order. If the execution is carried out on a network of several computers, there may be some loss or disorder. It will depend on many factors so any scenario is possible.
4. Create a server that collects the numbers that are sent from the client and that returns the sum at the moment of sending a zero.
5. Create a server that returns the European and Chinese zodiac signs from the date of birth sent by the client.

## 2. Verification

---

1. Which of the following internet protocols corresponds to the internet layer?
  - a) `TCP`.
  - b) `UDP`.
  - e) `HTTP`.
  - d) `IP`.
2. What feature is not specific to the `TCP` protocol?
  - a) It is connection oriented.
  - b) Ensures that packages arrive in an orderly manner.
  - c) Each packet can reach its destination by a different route.
  - d) Ensures that all packages are delivered to the recipient.
3. What is the `UDP` protocol most appropriate for?
  - a) For the transfer of web pages.
  - b) For voice communications.
  - c) To transfer files.
  - d) To send emails.
4. Which method of the `InetAddress` class provides the `IP` address in text mode?
  - a) `getByName`.
  - b) `getHostAddress`.
  - c) `getHostName`.
  - d) `getByAddress`.
5. Which method of the `ServerSocket` class is waiting to receive requests?
  - a) `accept`.
  - b) `bind`.
  - e) `close`.
  - d) `isBound`.
6. Which method of the `Socket` class provides a read stream?
  - a) `connect`.
  - b) `bind`.
  - e) `getInputStream`.
  - d) `getOutputStream`.
7. Which `DatagramSocket` method allows you to send a datagram?
  - a) `flush`.
  - b) `send`.
  - e) `getOutputStream`.
  - d) `connect`.
8. What information must be provided to a server socket?
  - a) The IP address.
  - b) The name of the host.
  - c) The request listening port.
  - d) The IP address of the client.

9. What information must be provided to a client socket?
- a) The IP address or the name of the server.
  - b) The IP address or name of the client.
  - e) The communication port used by the client.
  - d) The number of admitted threads.
10. What is the first step in creating a TCP socket server?
- a) Tell the server socket to wait for requests.
  - b) Accept the establishment of the connection.
  - c) Create a server-type socket associated with an address and a port.
  - d) Exchange data with the client.

## 3. Application activities

---

1. List the layers of the TCP/IP model.
2. Tell how IPv4 and IPv6 addresses are different.
3. Explain how a message is different from a packet in network communications.
4. Make a description of the datagram concept.
5. Find out which port MySQL database servers use by default.
6. Given a URL, find out the IP using the methods of the `InetAddress` class.
7. Given an IP, find out the URL of the host using the methods of the `InetAddress` class.
8. Explain the differences between TCP type sockets and UDP type sockets.
9. Explain why the web uses the TCP protocol and cannot use UDP.
10. List the steps to follow to create a TCP socket server.
11. List the steps to follow to create a UDP socket client.
12. Describe why a web server must have multiple threads to service client requests.
13. Explain why you would choose to use UDP as the protocol to develop a network game.

## 4. Extension

---

1. Deepens the knowledge of the open systems interconnection model (OSI). Look for information regarding the appearance of IPv6 addresses. Find out when they appeared and began to be used. Find out if they are 100% implemented.
2. Confirm that the following programming languages support TCP socket programming: Java, Python, PHP, C++, and Kotlin.
3. UDP sockets use datagrams and datagrams have a size limit. Find out what the minimum and maximum size is.
4. The Java `MulticastSocket` class is a subclass of `DatagramSocket`. Find out what it is for.
5. `WebSockets` allow real-time two-way communication between a server and a web browser. Deepen your knowledge of this technology.



## 5. Information sources

---

- [Wikipedia](#)
- [Programación de servicios y procesos - FERNANDO PANIAGUA MARTÍN \[Paraninfo\]](#)
- [Programación de Servicios y Procesos - ALBERTO SÁNCHEZ CAMPOS \[Ra-ma\]](#)
- [Programación de Servicios y Procesos - M<sup>a</sup> JESÚS RAMOS MARTÍN - \[Garceta\] \(1<sup>a</sup> y 2<sup>a</sup> Edición\)](#)
- [Programación de servicios y procesos - CARLOS ALBERTO CORTIJO BON \[Síntesis\]](#)
- [Programació de serveis i processos - JOAR ARNEDO MORENO, JOSEP CAÑELLAS BORNAS i JOSÉ ANTONIO LEO MEGÍAS \[IOC\]](#)
- GitHub repositories:
  - <https://github.com/ajcpro/psp>
  - <https://oscarmaestre.github.io/servicios/index.html>
  - <https://github.com/juanro49/DAM/tree/master/DAM2/PSP>
  - [https://github.com/pablohs1986/dam\\_psp2021](https://github.com/pablohs1986/dam_psp2021)
  - <https://github.com/Perju/DAM>
  - <https://github.com/eldiegoch/DAM>
  - <https://github.com/eldiegoch/2dam-ppp-public>
  - <https://github.com/franlu/DAM-PSP>
  - <https://github.com/ProgProcesosYServicios>
  - <https://github.com/joseluisgs>
  - [https://github.com/oscarovillo/dam2\\_2122](https://github.com/oscarovillo/dam2_2122)
  - [https://github.com/PacoPortillo/DAM\\_PSP\\_Tarea02\\_La-Cena-de-los-Filosofos](https://github.com/PacoPortillo/DAM_PSP_Tarea02_La-Cena-de-los-Filosofos)