

PROGRAMACIÓN DE SERVICIOS Y PROCESOS
TÉCNICO EN DESARROLLO DE APLICACIONES
MULTIPLATAFORMA

Protocolos nivel de aplicación

ÍNDICE

| | |
|---|----|
| / 1. Introducción y contextualización práctica | 3 |
| / 2. Introducción a los servicios en red | 4 |
| / 3. Protocolos a nivel de aplicación | 5 |
| / 4. Caso práctico 1: “¿Por dónde empezar?” | 5 |
| / 5. El protocolo DNS | 6 |
| / 6. El protocolo FTP | 7 |
| / 7. Los protocolos SMTP, POP3 e IMAP | 8 |
| / 8. Caso práctico 2: “No reinventar la rueda” | 8 |
| / 9. El protocolo HTTP | 9 |
| / 10. Resumen y resolución del caso práctico de la unidad | 10 |
| / 11. Bibliografía | 11 |

OBJETIVOS



Comprender el concepto de protocolo.

Conocer la capa de aplicación.

Conocer los protocolos DNS, FTP, SMTP, POP3 y HTTP.



/ 1. Introducción y contextualización práctica

En esta unidad, vamos a profundizar en la capa de aplicación del modelo OSI.

Vamos a recordar qué es un protocolo y vamos a estudiar algunos de los protocolos que se usan en comunicaciones y que actúan en la capa de aplicación.

Concretamente, los protocolos a los que nos referimos son el protocolo DNS, el FTP, el SMTP, el POP3 y el HTTP.

Vamos a analizar qué son y para qué sirve cada uno de ellos.

Escucha el siguiente audio, en el que planteamos el caso práctico que iremos resolviendo a lo largo de esta unidad.



Fig. 1. Administración de servidores de aplicaciones web



Audio intro. "Mejorando aplicaciones en red"

<https://bit.ly/34SD9zx>





/ 2. Introducción a los servicios en red

Para programar aplicaciones que se comuniquen mediante el envío y recepción de mensajes por red, necesitaremos una red de ordenadores interconectados entre sí. Esta la definimos formalmente como un sistema de telecomunicaciones interconectado entre sí y que tienen la finalidad de compartir información o recursos.

Aprovechando esta distribución de datos, se podrá hacer un mejor uso de ellos por parte de todos los usuarios que se encuentren dentro de la red informática, es decir, se va a mejorar, en gran medida, el rendimiento y la eficacia del sistema.

La utilización de redes de ordenadores presenta las siguientes ventajas:

- Reduce los costes tanto en *hardware* como en *software*.
- Podremos crear grupos de trabajo, los cuales nos ayudarán a tener organizados a los usuarios.
- Mejora de forma notable la administración, tanto de los equipos físicos como de las aplicaciones.
- Se mejora la integridad de los datos que hay en el sistema, que será mayor cuantos más nodos tenga el mismo.
- Lo mismo para la seguridad en los datos.
- Se facilita la comunicación.

Todo esto se debe a los servicios en red que estudiaremos en la siguiente unidad. No obstante, podemos ir adelantando que estos se van a clasificar en:

- **Servicios de administración/configuración:** Gracias a estos servicios, se va a facilitar tanto la administración como la gestión de configuración de los equipos. Un ejemplo son los servicios DHCP y DNS.
- **Servicios de acceso remoto:** Con ellos, podremos gestionar las conexiones de los usuarios a nuestra red desde lugares remotos. Un ejemplo son los servicios Telnet y SSH.
- **Servicios de ficheros:** Gracias a estos servicios, podremos ofrecer grandes cantidades de almacenamiento. Un ejemplo es el servicio FTP.
- **Servicios de impresión:** Nos facilitan la opción de imprimir documentos de forma remota.
- **Servicios de información:** Nos permiten obtener ficheros en función de su contenido. Un ejemplo es el servicio HTTP.
- **Servicios de comunicación:** Gracias a estos servicios, los usuarios podrán comunicarse a través de mensajes. Un ejemplo es el servicio SMTP.



Audio 1. "Definición de servicio en red"

<https://bit.ly/2Gd9DKp>





/ 3. Protocolos a nivel de aplicación

Actualmente, el conjunto de protocolos más importante lo conforman los englobados en el modelo TCP/IP. Sobre este modelo están basadas todas y cada una de las comunicaciones que realizamos cotidianamente.

Vamos a destacar, de entre todas las capas que conforman el modelo TCP/IP, la capa de Aplicación, que, si recordamos, es la capa que está más arriba dentro de este modelo. La capa de Aplicación es la que contiene absolutamente todos los protocolos que van a estar relacionados con los servicios en red.

En esta capa, se van a definir los protocolos que se van a utilizar por todas las aplicaciones que desarrollemos que necesiten intercambiar datos.

Cabe destacar que el número de protocolos que se encuentra en la capa de Aplicación aumenta continuamente, ya que no dejan de aparecer nuevos servicios que demandan los usuarios y que deben basarse en algún protocolo.

Algunos de los protocolos más importantes que nos vamos a encontrar dentro de la capa de Aplicación del modelo TCP/IP son los siguientes:

- **FTP:** Este protocolo se encarga de la transferencia de ficheros. Su nombre lo conforman las siglas de Protocolo de Transferencia de Ficheros (*File Transfer Protocol*).
- **Telnet:** Este protocolo nos permitirá poder acceder de forma remota a un ordenador y manejarlo.
- **SMTP:** Mediante el que podremos enviar correos electrónicos.
- **HTTP:** Es el que nos habilita la navegación por Internet. Es el usado por todos los navegadores web.
- **SSH:** Nos va a permitir una gestión segura de forma remota de otro ordenador. Es la versión mejorada del Telnet.
- **NNTP:** Con él podremos realizar el envío de noticias por la red. Su nombre lo conforman las siglas de Protocolo de Transferencia de Noticias (*Network News Transport Protocol*).
- **IRC:** Con el que podremos chatear vía Internet. Es el protocolo que siguen los chats de Internet.
- **DNS:** Este protocolo nos permitirá resolver direcciones de red.



Video 1. "Comunicación entre aplicaciones"
<https://bit.ly/3hPptZR>



/ 4. Caso práctico 1: "¿Por dónde empezar?"

Planteamiento. Pilar y José están estudiando algunos de los protocolos que existen en la capa de aplicación, los cuales van a dar servicio a múltiples aplicaciones que más tarde tendremos en nuestro sistema operativo instaladas.

«Seguramente, debemos implementar todos estos protocolos con los *Sockets* que ya hemos visto, no va a ser nada sencillo», le dice preocupado José a Pilar. Ella le responde que espera que no sea así, que seguramente habrá implementaciones de los protocolos ya hechas.



Nudo. ¿Crees que Pilar está en lo cierto y para programar alguna funcionalidad de estos protocolos ya existen bibliotecas predefinidas en Java?

Desenlace. Cuando estamos empezando a estudiar algún tema en concreto, es muy normal pensar en las implementaciones que pudiera haber elaboradas sobre ello.

El caso de las aplicaciones en red no es ninguna excepción y, si no albergamos mucho conocimiento sobre esto porque estamos empezando, vamos a pensar como nuestro amigo José: que estos servicios los vamos a tener que implementar desde cero utilizando *Sockets*.

En primer lugar, cabe destacar que se podría hacer, si se desea, una implementación desde cero utilizando *Sockets*, pero va a ser una tarea bastante ardua, ya que estos protocolos no son sencillos.

No obstante, como podemos imaginar, existen bibliotecas ya predefinidas en la JDK de Java que nos van a permitir trabajar con estos protocolos, facilitándonos el trabajo, en gran medida.

Un ejemplo puede ser la biblioteca [Proyect Lombok](#), la cual automatiza el código generado de los POJO.

También es importante tener en cuenta que, probablemente, podremos encontrar bibliotecas de terceros que trabajen con estos protocolos y que, seguramente, serán más sencillas de utilizar.

Con respecto a este tipo de bibliotecas, podemos destacar los controladores que nos ofrecen bases de datos como MySQL para poner conectar nuestros proyectos a ellas.



Fig. 2. Las bibliotecas y su potencial.

/ 5. El protocolo DNS

Todos los dispositivos, ya sean ordenadores, portátiles, smartphones, tables, videoconsolas, etc., que se puedan conectar a una red TCP/IP se van a identificar mediante una dirección IP del estilo *192.168.1.1*.

Las direcciones IP que se utilizan normalmente son del formato IPv4, las cuales se componen de cuatro números, que estarán comprendidos en el intervalo de 0 a 255, separados por un punto unos de los otros.

Los ordenadores pueden utilizar estos valores muy fácilmente para comunicarse entre los diferentes nodos, aunque para los humanos puede ser bastante complicado acordarse de todas las IP de los nodos o hosts a los que queremos acceder.

El servicio DNS, cuyas siglas hacen referencia a Sistema de Nombre de Dominio (*Domain Name System*), se encarga de resolver el problema de identificar a un nodo mediante una dirección IP. Este servicio nos va a permitir asignar nombres de dominio a un nodo, por ejemplo www.google.es, lo cual, para los humanos, es más fácil de recordar.

El servicio DNS se podrá utilizar en cualquier dispositivo que se conecte a la red.

El objeto principal del servicio DNS es el de traducir las direcciones IP a nombres de dominio y viceversa en cada uno de los dispositivos que están conectados a la red, pudiendo identificarlos de una forma mucho más sencilla.


Esto no quiere decir que tengamos que hacerlo así obligatoriamente. Si queremos acceder a los nodos de nuestra red mediante direcciones IP, podremos hacerlo sin ningún problema.



Algunas de las principales ventajas que nos ofrece el servicio DNS son las siguientes:


- Posibilita que varios nombres de dominio compartan una misma dirección IP (por ejemplo, si se tienen distintas marcas comerciales y se desarrollan distintas webs, pero se desea que todas apunten al mismo servidor).
- Permitirá que varias IP estén compartidas por varios nombres de dominio.
- Un nodo de la red podrá cambiar de nombre de dominio sin tener que cambiar de dirección IP.

El servidor DNS va a utilizar una base de datos distribuida, que es la encargada de almacenar toda la información asociada a nombres de dominio en redes como Internet.



Video 2. "Conexión, comunicación y desconexión"

<https://bit.ly/32RdDbS>



/ 6. El protocolo FTP

Hoy en día, una de las funcionalidades más utilizadas en la red es la de poder transferir ficheros, sean del tamaño que sean, desde varios ordenadores, en una red local o desde Internet.

Para proporcionarnos este servicio surgió el protocolo FTP, que proporcionará las funcionalidades necesarias para definir un estándar en la transferencia de ficheros en las redes TCP/IP.

Podemos decir que son dos los principios fundamentales del protocolo FTP:

- Nos va a permitir intercambiar ficheros entre ordenadores remotos que estén interconectados mediante una red.
- Permite transferir los ficheros a una velocidad bastante alta.

La gran desventaja de este protocolo es que transmite toda la información en formato de texto plano. Esto significa que la información transmitida no está codificada, por lo que, si alguien realiza una conexión intermedia, podrá obtener los ficheros transmitidos de un extremo a otro. La transferencia se realiza en texto plano para proporcionar una mayor velocidad de transferencia de información, a costa de una pobre seguridad.

El problema de la seguridad será solucionado gracias a la encriptación de la información a través del protocolo SFTP, Protocolo de Transferencia de Ficheros Seguro, usado junto al protocolo SSH.

El protocolo FTP usa el puerto 20 para la transmisión de datos y el 21 para la transferencia de órdenes.

Algunas de las características del protocolo FTP son:

- Permite conectar usuarios en remoto al servidor FTP.
- Hay un límite en el acceso al sistema de archivos mediante un sistema de privilegios de los usuarios.
- Tiene dos modos de conexión: el modo activo y el pasivo. En el modo activo habrá dos conexiones distintas, mientras que en el modo pasivo no.

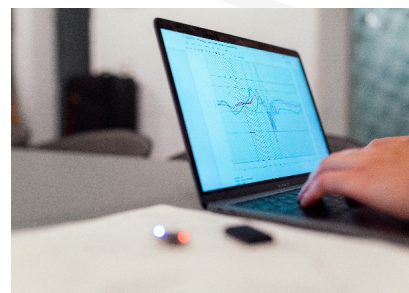


Fig. 3. FTP: Velocidad vs. Seguridad.

/ 7. Los protocolos SMTP, POP3 e IMAP

El correo electrónico se ha convertido en una parte fundamental e irremplazable de nuestra forma de trabajar en la actualidad, hasta el punto en el que no podemos imaginar nuestro día a día sin él.

El servicio de correo electrónico permite tanto enviar como recibir mensajes con o sin archivos de una forma muy rápida a través de Internet.

El servicio de la capa de Aplicación que nos va a permitir usar el correo electrónico es el protocolo SMTP, cuyas siglas significan Protocolo para Transferencia de Correo Simple (*Simple Mail Transfer Protocol*), el cual sigue el modelo Cliente/Servidor. Por lo tanto, deberemos tener un servidor de correo electrónico y un cliente de correo electrónico para poder usarlo correctamente.

El servidor de correo electrónico creará una serie de cuentas para los usuarios, que tendrán lo que se denomina un buzón, donde estarán almacenados todos sus correos correspondientes.

Los clientes serán los encargados de descargar y elaborar los correos electrónicos.

El protocolo SMTP utiliza el puerto 25 y es el encargado de realizar el transporte del correo desde la máquina del usuario hasta el servidor, que lo almacenará para que el destinatario del mismo pueda acceder a él.

Cuando el destinatario desee ingresar en su cuenta de correo electrónico, tendrá dos opciones para acceder a los mensajes: descargarlos en su máquina local mediante el protocolo POP3 o consultarlos directamente en el servidor, mediante el protocolo IMAP.

Estos protocolos nos permitirán no solo enviar texto en nuestros correos electrónicos, sino cualquier tipo de documento digitalizado.

El protocolo SMTP utiliza una serie de comandos para poder realizar la transferencia del correo. Algunos de ellos son:

- **HELO:** Comando usado para abrir una sesión con el servidor.
- **MAIL FROM:** Comando usado para indicar quién es el emisor del correo.

```
// Convertimos el array json
desde i = 0 hasta json.tamaño
dni = arrayjson.getElemento(i).obtenerDNI()
nombre = arrayjson.getElemento(i).obtenerNombre()
apellidos = arrayjson.getElemento(i).obtenerApelli
persona = Persona(dni, nombre, apellidos)
arraypersona.agregar(persona)

// Mostramos el array
desde i = 0 hasta arraypersona.tamaño
persona = arraypersona.getElemento(i)
mostrar(persona)
```

Fig. 4. Procesamiento del correo.

/ 8. Caso práctico 2: “No reinventar la rueda”

Planteamiento: De nuevo, nuestros amigos Pilar y José ya casi tienen resuelta una nueva aplicación que les han encargado.

No obstante, los clientes se han puesto en contacto con ellos para realizar un cambio de última hora. Necesitan un navegador web integrado en su aplicación, y es bastante importante que lo integren.

Ante esta repentina petición, nuestros amigos se preguntan cómo pueden resolverlo: «¿y si nos creamos nuestro propio navegador web?, seguro que hay alguna forma», le dice Pilar a José.

Nudo: ¿Qué piensas al respecto? ¿Crees que deberían llevar a cabo la propuesta de Pilar y realizar un navegador web desde cero?



Desenlace: Uno de los pasos más importantes cuando se nos presenta un nuevo problema es investigar un poco sobre él y no aventurarse a desarrollar código sin haber estudiado diversas opciones previamente.

En este ejemplo en concreto, nuestros amigos tienen que integrar un navegador web en una aplicación, cosa que se podría hacer desde cero, pero que implicaría mucho tiempo y esfuerzo.

Una forma más simple de integrar un navegador web en una aplicación sería utilizando la interfaz gráfica, y como el 99,9% de las aplicaciones comerciales la utilizan, no deberíamos tener problema.

Si recordamos lo estudiamos en la asignatura de *Programación*, en la [biblioteca gráfica SWING](#), existía un elemento que era un navegador web, que podíamos integrar directamente en nuestras aplicaciones.

Una vez hecho esto, incluso podíamos forzar ciertos cambios en el navegador, como utilizar navegación segura, de una forma muy sencilla con las propiedades del mismo.

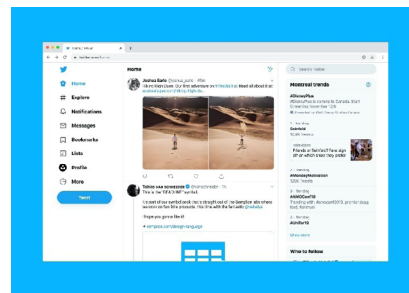


Fig. 5. Navegador web.

/ 9. El protocolo HTTP

Quizás el protocolo más famoso de todos los que vamos a ver en esta unidad es el protocolo que nos va a permitir la navegación por Internet.

El protocolo HTTP, cuyas siglas significan Protocolo de Transferencia de Hipertexto (*HyperText Transfer Protocol*), es el protocolo encargado de que podamos navegar por Internet de forma correcta. Está compuesto por una serie de normas que posibilitan una comunicación y transferencia de información entre cliente y servidor.

La información que transfiere este protocolo son las páginas HTML.

El protocolo HTTP se encarga de definir toda la sintaxis de comunicación que van a usar tanto el cliente como el servidor, siendo algunas de las reglas más importantes las siguientes:

- HTTP sigue el modelo de petición-respuesta que aplica al servidor y al cliente.
- El puerto que utiliza es el 80, aunque ofrece la posibilidad de cambiarlo.
- A cliente se le denomina como agente del usuario, o *user agent* en inglés.
- Toda la información que se transmite se conoce como recursos y están identificados mediante una URL del tipo <http://www.google.es>.
- Los recursos también pueden ser ficheros, una consulta de una base de datos, un resultado de una operación realizada por un programa, etc.

El protocolo HTTP no tiene estado. Con esto, se refiere a que no va a recordar absolutamente nada de conexiones anteriores.



Básicamente, el protocolo HTTP funciona de la siguiente forma:

- El usuario accede a una URL, pudiendo indicar el puerto en la misma.
- El equipo cliente descompondrá la información de la URL, diferenciando todas sus partes, como el nombre de dominio, la IP, el puerto, etc.
- El cliente establece una conexión con el servidor.
- El servidor contesta al cliente y envía el código HTML.
- El cliente visualiza el HTML y cierra la conexión.

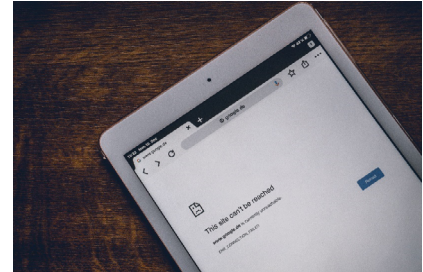


Fig. 6. Navegando por Internet.

/ 10. Resumen y resolución del caso práctico de la unidad

En esta unidad, hemos estudiado un poco más en profundidad la **capa de aplicación** del modelo OSI.

Hemos recordado qué es un **protocolo** y hemos visto algunos de los protocolos que más se usan en comunicaciones y que actúan en la capa de aplicación.

Los protocolos que hemos conocido han sido el protocolo **DNS**, que actúan en la resolución de nombres de dominio; el protocolo **FTP**, usado para la transferencia de ficheros por la red; el protocolo **SMTP**, para la gestión del correo electrónico; el protocolo **POP3**, también para la gestión del correo electrónico, y el protocolo HTTP, usado para el transporte de hipertexto por la red, o lo que es lo mismo, para poder navegar por Internet.

En cada caso, hemos analizado qué son y para qué sirven.

Resolución del caso práctico de la unidad

Una de las posibilidades que podemos encontrar cuando nos dedicamos al mundo de la programación, sea cual sea nuestra especialidad, serán aplicaciones que haya que mejorar o ampliar. Aunque también, por supuesto, tendremos la oportunidad de desarrollar aplicaciones desde el principio.

Esto también se puede dar en las aplicaciones en red, y el procedimiento a seguir es el mismo para todas ellas: básicamente, tendremos que estudiar y analizar el código para ver cómo funciona, con el objetivo de analizar posibles fuentes de error o de bloques de códigos ineficientes, para arreglarlos y mejorar la aplicación.

En el caso que concierne a nuestros amigos, podríamos implementar un control de acceso de los usuarios, para que solo puedan realizar peticiones en una cantidad predeterminada que sepamos que no va a sobrecargar la red. Para esto, podríamos implementar en el servidor un sistema de hilos muy simple, en el que comprobemos, mediante un contador, si se han sobrepasado la cantidad de usuarios permitidos entre un cliente; en ese caso, lo dejaríamos en espera y, una vez salga otro cliente, disminuiríamos ese contador y le daríamos una señal al cliente que se encontraba en espera para que entre.



Fig. 7. Accesos concurrentes.



/ 11. Bibliografía

- Colaboradores de Wikipedia. (2019, 8 diciembre). *Capa de aplicación*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Capa_de_aplicaci%C3%B3n
- Colaboradores de Wikipedia. (2020a, marzo 17). *Protocolo de transferencia de archivos*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_archivos
- Colaboradores de Wikipedia. (2020a, mayo 8). *Protocolo para transferencia simple de correo*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Protocolo_para_transferencia_simple_de_correo
- Colaboradores de Wikipedia. (2020a, mayo 11). *Protocolo de configuración dinámica de host*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Protocolo_de_configuraci%C3%B3n_din%C3%A1mica_de_host
- Colaboradores de Wikipedia. (2020a, junio 12). *Sistema de nombres de dominio*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Sistema_de_nombres_de dominio
- Colaboradores de Wikipedia. (2020, 12 julio). *Protocolo de transferencia de hipertexto*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto
- Gómez, O. (2016). *Programación de Servicios y Procesos Versión 2.1*. https://github.com/OscarMaestre/ServiciosProcesos/blob/master/_build/latex/ServiciosProcesos.pdf
- Sánchez, J. M. y Campos, A. S. (2014). *Programación de servicios y procesos*. Alianza Editorial.

C
A
D
E
M