Consigue Empleo o Prácticas
Matricúlate en IMF y accede sin coste a nuestro servicio de Desarrollo Profesional con más de 7.000 ofertas de empleo y prácticas al mes.

IMF Smart Education
¿Quieres conocer todos los servicios?

T5: servicios y protocolos de aplicación en Internet

Introducción a las aplicaciones de red

Interacción cliente servidor

Servidor: **Siempre funcionando** en una **IP permanente y pública**, accesible **sin abrir puertos ni NAT**. Se despliegan en **granjas** gestionadas por un hipervisor que lanza instancias según las necesidades.

Clientes: Funciona **intermitentemente**, pueden tener **IP dinámica y privada**, se **comunican con el servidor** para hacerle solicitudes.

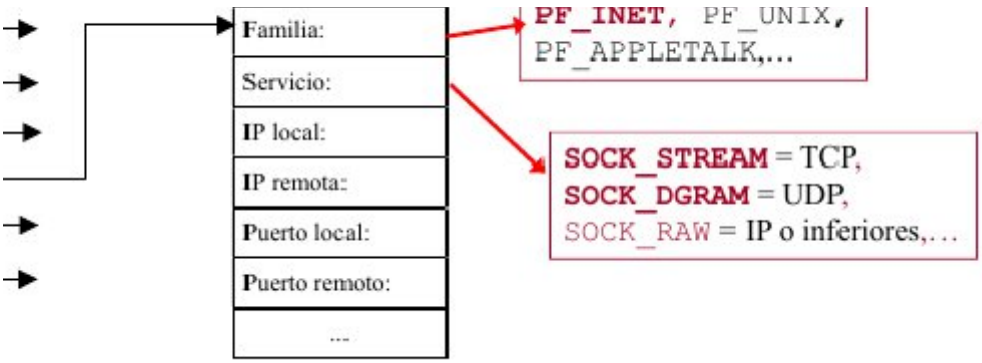
Algunas alternativas:

- P2P
- Publicador/subscriptor

Interfaz socket

El cliente manda solicitudes desde su proceso (controlado por el desarrollador de aplicaciones) a un **socket** que sirve de **intermediario con la capa de transporte**, que es controlada por el SO. Para recibir mensajes, un proceso debe tener un **identificador (IP+puerto).**

Socket: Es muy parecido al descriptor de fichero. Igual que para escribir y leer en un fichero necesito una variable para interactuar con él, necesito abrir un socket a través del cual pueda enviar y recibir mensajes y luego cerrarlo. Es un **descriptor de la transmisión**. Es un puntero a una estructura:



- **IP:** Para la conexión.
- **Puerto:** Para la conexión.
- **Familia:** INET (constante simbólica que el socket va a escribir una **conexión de internet**) antes estaba la de apple o la de unix, ahora la dominante es inet.
- **Servicio**: Puede decirnos si es tcp, udp o si va a usar datagramas IP o inferiores.

WUOLAH
Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Para estimar retardos de cola en un servidor (o un router) se utiliza la **teoría de colas**. Por si acaso, dejo aquí la fórmla del retardo de cola pero dudo que los ejercicios sean de eso:

$$R = \frac{\lambda \cdot (T_s)^2}{1 - \lambda \cdot T_s}$$

Donde Ts es el valor esperado del tiempo de servicio y λ es el valor esperado de la tasa de llegada de solicitudes.

Protocolos

Un protocolo es una serie de procedimientos que definen:

- El **tipo de servicio** (orientado o no a conexión, confirmado o no).
- **El tipo de mensaje** (request o response).
- **La sintaxis** (campos del mensaje, texto o binario).
- **Semántica** (significado de los campos).
- **Las reglas** (cuándo y para qué son las solicitudes y respuestas de mensajes)

Tipos de protocolos

Según **licencia**:

- Dominio público: Cualquiera puede utilizarlos, código libre, libre comercialización y modificación (HTTP).
- Propietarios: Solo sus propietarios conocen el código, no se pueden utilizar ni modificar libremente ni comercializar sus variaciones (Skype).

Según **ubicación del controlador en el sistema** de comunicación:

- In-band: Usan la **misma conexión para datos y control**, menos complejo y más eficiente. Pueden ser menos seguros.
- Out-of-band: Más seguros, **separan controlador y datos**, evita ataques.

Según **cómo** un servidor **mantiene la información** sobre las sesiones de cliente **entre consultas**:

- Stateless: **Cada solicitud** del cliente **es independiente** y no depende de ninguna anterior o futura.
- State-full: Mantiene un “estado” que **recuerda las solicitudes previas** y personaliza/adapta la respuesta.

Según cómo un servidor **mantiene la información** sobre las sesiones de cliente **después de** que el cliente ha **finalizado su conexión**:

- Persistentes: el servidor **guarda toda o parte de la información de la sesión del cliente**, así que puedes recuperarlo tal como lo dejaste.
- No-persistentes: **El servidor no guarda ninguna información sobre las sesiones del cliente** después de cerrar la conexión.

Se suelen hacer los protocolos con **cabeceras fijas + trozos (obligatorios o no)**

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

Los trozos pueden tener **parámetros fijos o variables** en **formato TLV** (tipo, longitud y valor del parámetro).

Características de las aplicaciones de red

1. Tolerancia a errores: **Algunas** apps (audio) **pueden tolerar pérdida** de datos; **otras** (HTTP) **no**.
2. Exigencia de requisitos temporales: Las apps **inelásticas** (juegos) **requieren** retardo (**latencia**) reducido, **otras no**.
3. Demanda de ancho de banda (tasa de transmisión o throughput): Algunas requieren **envío de datos a una tasa determinada** (codec de vídeo), otras no.
4. Nivel de seguridad: Los requisitos de seguridad son **variables** (encriptación, autenticación...)

Conclusión: Las distintas aplicaciones tienen requisitos HETEROGÉNEOS (diferentes)

Protocolos de transporte

TCP	UDP
Orientado a conexión Transporte fiable, control de errores Control de flujo Control de congestión	No orientado a conexión Transporte no fiable Sin control de flujo Sin control de congestión

No garantizan **QoS** (calidad de servicio) porque **no** están acotados **el retardo ni sus fluctuaciones** ni la **probablidad de pérdidas** y **no** hay una **velocidad mínima** de transmisión. Tampoco tienen seguridad garantizada.

Servicio de Nombres de Dominio (DNS)

Traduce dominios a IPs y viceversa para encaminarnos. Trabaja en el **puerto 53**. Existe una jerarquía de dominios:

1. **Raíz** (.), gestionado por el ICANN. Esta misma entidad delega algunos TLD a regiones, que gestionan las limitaciones de lo que le cedan.

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

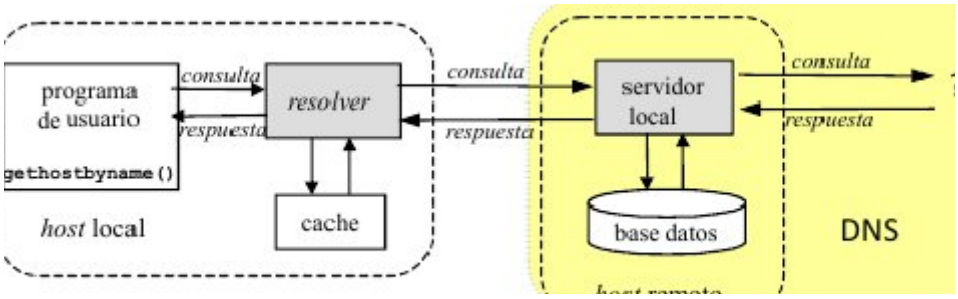
WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

2. **TLD** (Top level domains): .com .es .jp .edu ... Hay más de 1500.
3. **Local server** (www o ugr)

El objetivo: dado un dominio tener una IP y que no me entere (transparente), automático.
Proceso:

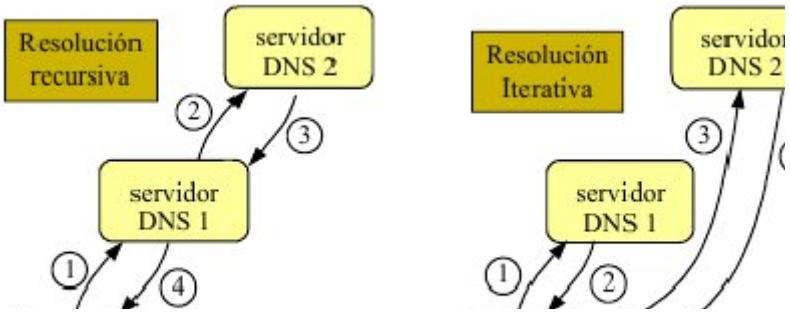
1. Consultar caché: El **host local le pasa la consulta al resolver**. El resolver lo único que hace es **consultar la caché**, por si en modo local alguien ha realizado la consulta antes y está guardada. **Si no**, entonces **el resolver se vuelve un cliente DNS.**
2. Preguntar DNS local: Le hace una consulta al servidor DNS local por TCP o UDP con el puerto 53. El servidor l**ocal tiene una porción del arbol de dominios**. Lo más **normal** es que **no sepa el nombre** de recurso. El **servidor local se convierte en cliente** y hace una **consulta al ROOT.**
3. Resolución: El **root puede o no saber** la respuesta. **Si no** lo sabe, **analiza sintácticamente, ve el TLP** y **traslada la consulta al servidor del TLP**. Así sucesivamente **hasta que llegamos a** un servidor que sí sabe la **respuesta**.



Tipos de consultas:

- **Recursivas**: el host y el usuario son partícipes en todo momento. Actualizas cachés pero gastas más recursos.
- **Iterativas**: El host consulta al servidor local, le dice al user que pregunte al root, el root dice que preguntes al servidor de japón, el de japón que preguntes a osaka etc.

Se pueden mezclar, normalmente host a server local es recursiva y el resto iterativas.



Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

Importante
Puedo eliminar la publi de este documento con 1 coin
¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

Gestión de Base de datos DNS (BIND)

Hay una base de datos distribuida llamada **BIND,** de la cual cada servidor DNS tiene una porción, donde se almacenan registros IP y dominio y otras cosas. Cada servidor es **responsable de una zona**. Por ejemplo, de .es, serían todos los dominios que estén debajo de .es (los que acaben en .es). **Cada zona** tiene una **autoridad** que se **puede delegar** (.es delega .ugr a la universidad, ya no necesita saber esos dominios).

En cada zona **hay más de un servidor**. Hay **primarios** (copia master de la base de datos) y **secundarios** (réplicas). Como todas las consultas van al root, hay un **cuello de botella,** así que existen **13 servidores root con múltiples réplicas con la misma IP** por el mundo. Nosotros, por ejemplo, tenemos una réplica del servidor F en Madrid.

Cuando un cliente solicita DNS, puede que este tenga:

- Respuesta con autoridad: Es su zona.
- Respuesta sin autoridad: Tiene el nombre en caché
- No conoce respuesta: pide a root.

Base de datos DNS

Se almacena en txt denominados **zone files** y cada uno tiene registros llamados **RR** (Resource Record). Cada zone file, determina un **TTL** (Time To Live), que es el tiempo de validez de los RR en caché.

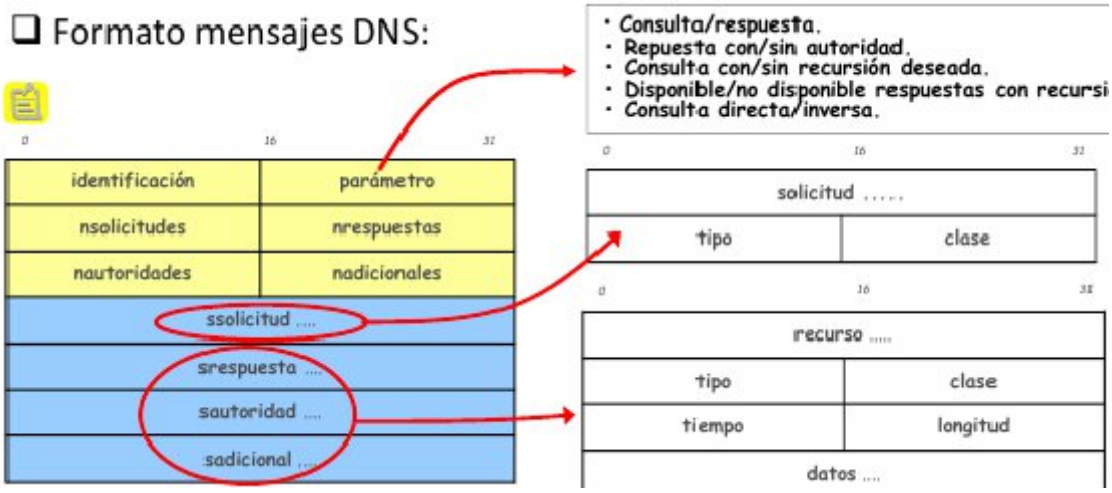
Un RR contiene 5 campos:

- **Nombre de dominio**
- **Clase**: (previsto para varios tipos, ya todo es INET así que) IN
- **Tipo**: puede especificar varias cosas, no solo una IP
    - SOA, start of authority dice quién es la autoridad para una zona.
    - A, la IPv4
    - MX, mail exchanger.
    - CNAME alias de dominio
    - HINFO info sobre el hardware y SO del sv
    - TXT texto libre
- **Valor**: el contenido dependiente de tipo.

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Formato mensaje DNS



Amarillo cabecera, azul solicitudes. Puedes decir siso de recursión o iteración en respuesta y solicitud. Lo normal es iterativo.

La navegación Web (HTTP)

Una **página web es un fichero formado por objetos** (html, imágenes, audio, vídeo). **Cada objeto** tiene su **URL**.

Según el protocolo que se utiliza, cambia el inicio de la url. Por ejemplo, Gopher tiene enlaces del tipo “gopher://...”, http “http://...” y así.

Las páginas se sirven con el **protocolo http** con el siguiente modelo **cliente servidor**:

- Cliente: Navegador, solicita páginas web y las muestra.
- Servidor: Envía objetos web según peticiones.

Las páginas, **según cómo se generan y muestran** los contenidos, pueden ser:

- Estáticas: El contenido **pregenerado** se almacena en el servidor de la web. El servidor sirve los datos al navegador tal cual los tiene almacenados.
- Dinámicas: El contenido **se genera en tiempo real**, procesa las bases de datos y genera una respuesta html personalizada.

Las páginas **dinámicas**, pueden proporcionar **contenido diferente**:

- Scripting en cliente: JavaScript, por ejemplo.
- Scripting en servidor: El código lo procesa el servidor. Python, por ejemplo. Se hacen incrustando etiquetas en la web que genera solicitudes al servidor una vez se accede a ellas.

Características

Usa **TCP** en el **puerto 80**. Es **stateless** (usa **cookies** [ficheros temporales de estado y su valor], el servidor no mantiene información de sesión). Es **in-band** (mismo canal para control,

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

usa cabeceras), **orientado a texto** (optimizado para mandar y recibir contenido basado en texto)

Las cookies **arreglan la falta de información de estado** entre transacciones. La primera vez que un usuario accede a un documento, el servidor le da una cookie que utilizará para operaciones poesteriores. Se **guarda en** el ordenador del **cliente**.

Tienen **dominio** (direcciones para las que es valida la cookie), **path** (URLs para las que sirve la cookie), **version** (versiones del modelo de cookies)**, expires** (si no se incluye se descartan al finalizar la sesión). Usan la cabecera **set-cookie** después de un **GET**. Después el cliente usará la cabecera **cookie**.

Tipos de servidores HTTP:

- No persistente (HTTP/v1.0): Un objeto por cada conexión TCP.
- Persistente (HTTP/v1.1): Múltiples objetos en una conexión TCP.

FTP sí necesita dos canales para las conexiones. Puerto 20 para el control y 21 para los ficheros.

Proceso de solicitud:

1. **Cliente** (navegador) **solicita** un **objeto** con su URL. Si no se especifica, te da index.html.
2. **Cliente consulta al resolver** DNS para la IP.
3. El cliente **abre una conexión TCP** con el puerto **80** de la IP.
4. Hace una **petición GET** <objeto>. Puede añadir **info adicional** (cookies, cabeceras).
5. El **servidor manda el objeto**. Es una conexión transparente y fiable por TCP.
6. **Si** es **persistente** se **siguen** pidiendo solicitudes GET.
7. Se **cierra TCP**.

Tipos de mensaje http

Request

Formato:

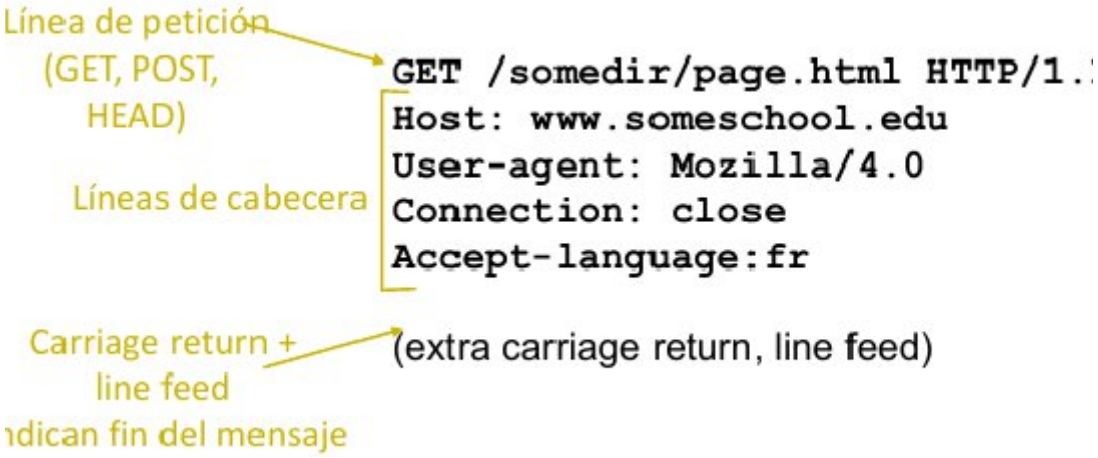
1. **Línea de petición**:
    a. GET, POST, PUT, DELETE... + nombre del objeto + HTTP/1.1 o HTTP/1.0
2. Info. de control con la **cabecera** (in band).
    a. Host, user agent, language...

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

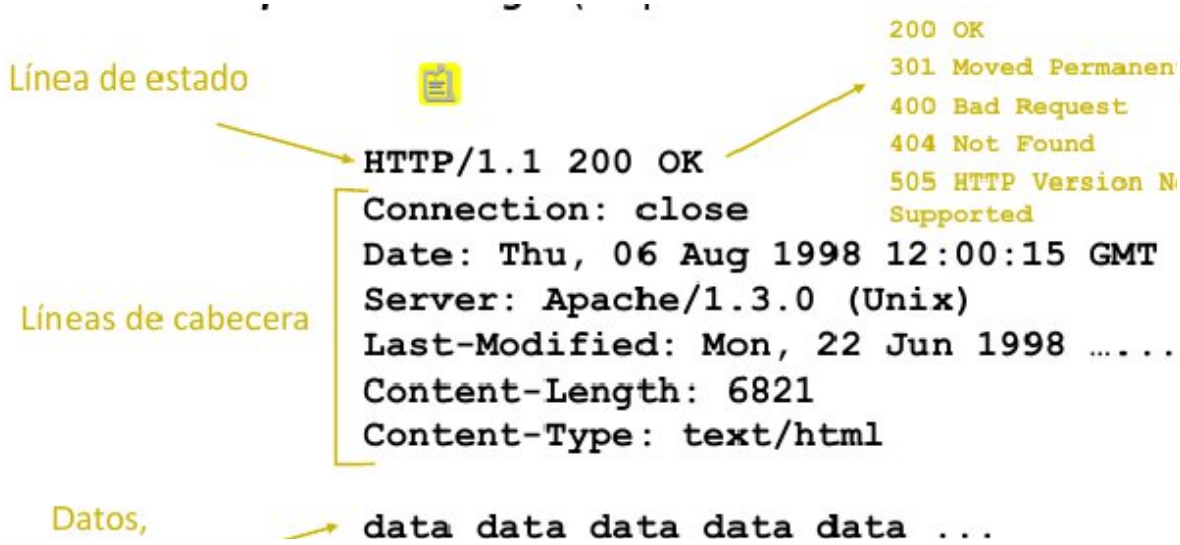
3. A veces algunas variables dependiendo del método



Response

Formato:

1. **Línea de estado**:
    a. HTTP/1.1 o HTTP/1.0 + código http (200, 404, 402...)
2. Info. de control con la **cabecera** (in band).
    a. Fecha, server, longitud, tipo de contenido...
3. Datos



- Métodos:
    - **OPTIONS**: info sobre opciones disponibles.
    - **GET**: solicitar recurso.
    - **HEAD**: GET pero solo devuelve cabeceras.
    - **POST**: se utiliza para crear nuevos recursos en el servidor.
    - **PUT**: actualiza recursos ya existentes en el servidor.

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

Consigue Empleo o Prácticas

Matricúlate en IMF y accede sin coste a nuestro servicio de Desarrollo Profesional con más de 7.000 ofertas de empleo y prácticas al mes.

IMF Smart Education

- **DELETE**: borrar URI (identificador de recurso).

- Códigos de respuesta:
  - **1xx** información.
  - **2xx** éxito.
  - **3xx** redirección a otra URL.
  - **4xx** error en la solicitud.
  - **5xx** error en el servidor o protocolo.
- Cabeceras comunes para peticiones y respuestas:
  - **Content-Type**: Descripción MIME (tipo de datos que están siendo enviados, por ej audio).
  - **Content-Length**: Longitud en bytes de datos enviados
  - **Content-Encoding**: Formato de codifiación de los datos (zip).
  - **Date**: fecha de la operación.
- Cabeceras solo para peticiones del cliente:
  - **Accept:** tipos MIME aceptados por cliente.
  - **Authorization:** Clave de acceso para recurso protegido.
  - **From:** Dirección de correo del usuario del cliente Web.
  - **If-Modified-Since**: Get condicionales con fecha.
  - **Referer:** Desde donde se ha activado un objeto web.
  - **User-agent:** Tipo y versión del cliente.
- Cabeceras para respuestas del servidor HTTP:
  - **Allow**: Comandos HTTP ocionales que se pueden aplicar sobre el objeto.
  - **Expires**: Fecha de expriación del objeto en caché.
  - **Last-modified:** Fecha local de modificación del objeto.

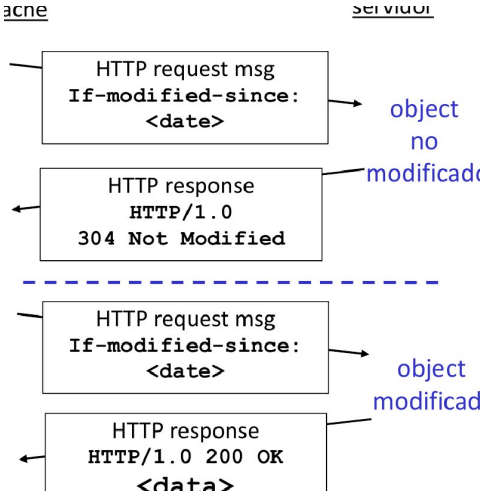
¿Quieres conocer todos los servicios?

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Web cache

Queremos reducir el tráfico, así que se le hacen solicitudes a la caché o al servidor proxy y si no está se solicita al servidor y se guarda en caché.



HTTP/1.1 200 OK
Date: Fri, 30 Oct 1998 13:19:41 GMT
Server: Apache/1.3.3 (Unix)
Cache-Control: max-age=3600
Expires: Fri, 30 Oct 1998 14:19:41 GMT
Last-Modified: Mon, 29 Jun 1998 02:28:12 GMT
ETag: "3e86-410-3596fbbc"
Content-Length: 1040

Acceso restringido

HTTP no es seguro, pero puedes restringir acceso con login y password. Impone un mecanismo de autenticación- Cuando hacemos un GET, no devuelve el objeto, nos manda un mensaje 4xx y un autenticate (una cadena de texto q es una zona de seguridad).

Lanza un popup de login y password y en la respuesta va la cabecera authorization y las credenciales que suelen ser user:password codificado en base 64

El Correo electrónico (SMTP/IMAP/POP3)

- **MUA**: (Mail user agent). Los ejecutables a través de los cuales el usuario lee, compone correos electrónicos (outlook, thunderbird...) Tienen GUI. Interaccionamos con el sistema.

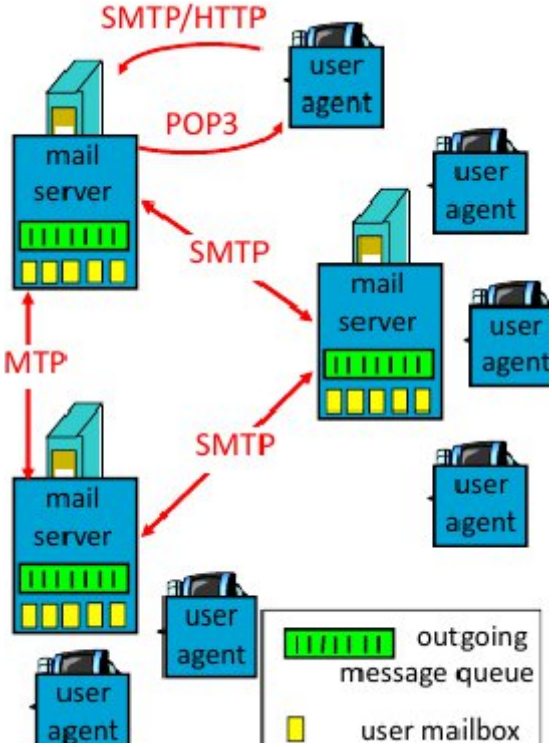
- **MTA**: (Main transfer agents). Intermediarios entre MUA y MUA. **Mail servers**. Se ejecutan permanentemente. Tiene el rol de servidor y cliente.

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

En todo envío de correo está SMTP. De MUA a MTA y también de MTA a MTA. Para interaccionar con nuestro buzón se usa otro protocolo. De MTA a MUA puede ser POP3, IMAP.



SMTP es el protocolo de envío. El cliente se ejecuta en MTA (server) o en MUA (user), mientras que el servidor solo en MTA, porque es el único que recibe el correo. Utiliza el **puerto 25** de TCP. Es **in-band** y **state-full**.

Fases:

1. Handshaking
2. Transferencia
3. Cierre

Inicialmente los mensajes eran ascii pero añadieron extensiones MIME para poder enviar imágenes, audio, vídeos...

Pasos de envío/recepción correo:

1. EL usuario compone el mail con su MUA.
2. Se envía con SMTP/HTTP el mensaje al MTA.
3. El MUA abre una conexión TCP con el MTA.
4. MUA manda SMTP por puerto 25
5. El MTA manda el mensaje en el mailbox destino
6. El destino invoca a su MUA para leer con POP3, IMAP o HTTP

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

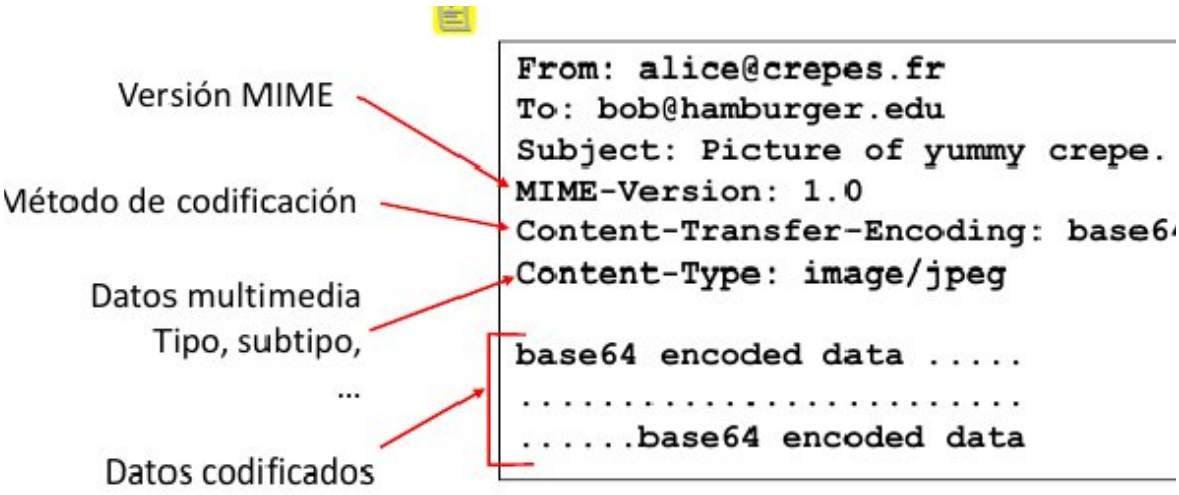
WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

SMTP no es seguro, puedes suplantar el origen.

MIME

Soportan texto con caracteres diferentes, adjuntos, mensajes con múltiples partes, etc. Tiene una **cabercera** MIME-Version. Indica que hay cabeceras mime y puedes poner el tipo de contenido etc etc.



Cabecera	Descripción
MIME-Version:	Identifica la version de MIME. Si no existe se consid que el mensaje es texto normal en inglés.
Content-Description:	Cadena de texto que describe el contenido. Esta cadena necesaria para que el destinatario sepa si de descodificar y leer el mensaje o no.
Content-Id:	Identificador único, usa el mismo formato que la cabec estándar Message-Id.
Content-Transfer-Encoding:	Indica la manera en que está envuelto el cuerpo

Hay varios tipos de contenido mime que se pueden adjuntar:

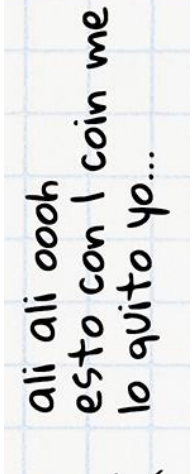
Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada

Importante
Puedo eliminar la publi de este documento con 1 coin
¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

Postscript	Documento imprimible PostScript.
Rfc822	Mensaje MIME RFC 822.
Partial	Mensaje dividido para su transmisión.
External-body	El mensaje mismo debe obtenerse de la red.
Mixed	Partes independientes en el orden especificado.
Alternative	Mismo mensaje en diferentes formatos.
Parallel	Las partes deben verse simultáneamente.
Digest	Cada parte es un mensaje RFC 822 completo.



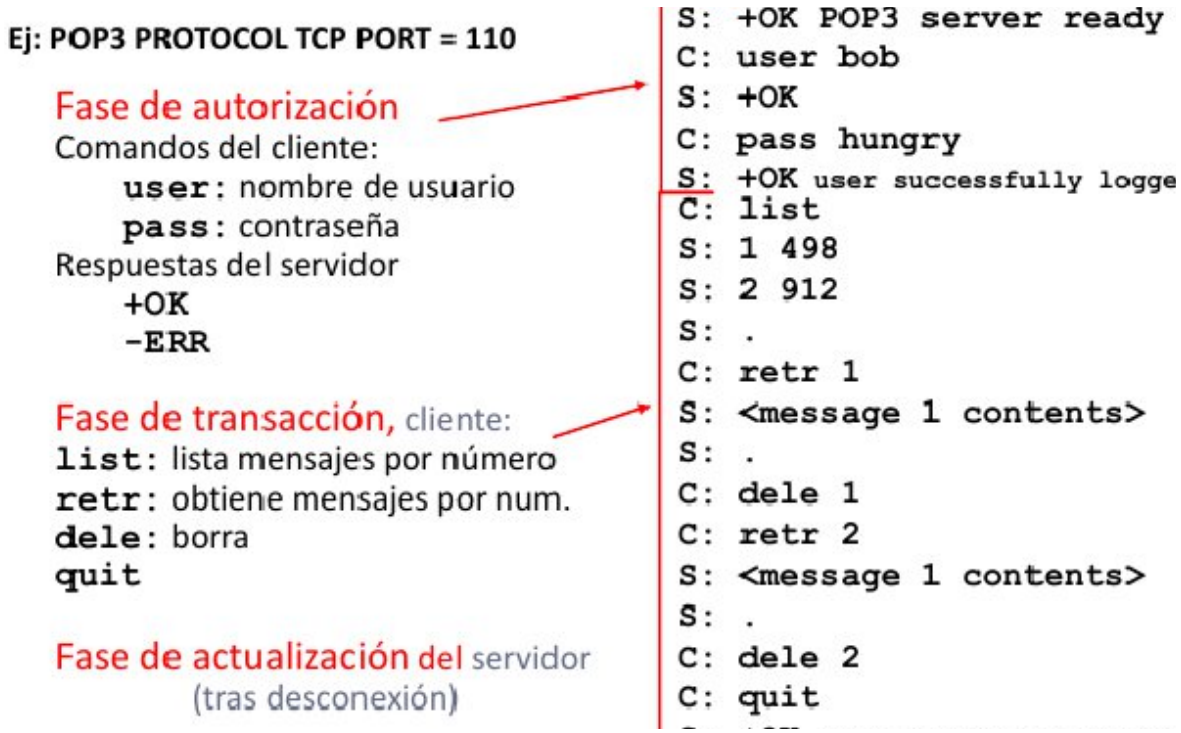
- Los **application** añadien formatos que requieren procesammiento externo.
  - **Octet-stream** son bytes no interpretados. Se copian a un archivo y se solicita nombre del archivo.
  - **Postscript** es el lenguaje de adobe. Es un lenguaje de programación tal cual y puede ejecutarse en un intérprete.
- Los **messages** permiten encapsular mensajes dentro de otros (para reenviar).
  - **Partial** divide un mensaje en trozos y los envía por separado.
  - **External-body** Para mensajes miy grandes, se da una dirección ftp y puede obtenerlo el receptor cuando quiera.
- **Multipart** tiene más de una parte el mensaje
  - **Mixed** que cada parte es diferente
  - **Alternative** Cada parte el mismo mensaje pero en medios distintos
  - **Parallel** las partes se ven simultáneamente
  - **Digest** cuando se juntan muchos mensajes en uno compuesto

POP3

Usa el **puerto 110** Tiene una autorización que no es muy segura con user y password. Luego hace una transacción y por último actualiza el servidor tras desconectarse.

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.



La **diferencia con IMAP** es que IMAP puede **organizar** en **directorios en** el **servidor** y **gestiona estados**, mientras que POP3 no. También permite **descarga parcial** de los mensajes. IMAP usa el puerto 143.

Aplicaciones multimedia

Requieren audio, vídeo, juegos, real-time a través de Internet.

Queremos **favorecer QoS** (quality of service) que es la capacidad para **ofrecer el rendimiento exigido por la aplicación**. IP es best effort, así que no garantiza nuestras exigencias.

Podemos tener aplicaciones con audio y vídeo almacenado (streaming) -> youtube, con flujo de audio y vídeo en vivo (twitch o radio), o audio y vídeo interactivo (skype, discord...)

Para dar soporte para aplicaciones que exigen calidad de servicio (qos) como las aplicaciones industriales, telefonía y juegos, se **añaden componentes a los routers**.

Clasifican el tráfico y le dan distintos tratamientos según el tipo.

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Plan Turbo - Eliminar los vídeos + 10 descargas sin publicidad por sólo 0,99€ / mes - Oferta limitada