



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

**Licenciatura en Ciencias de la
Computación**

Redes de Computadoras

Unidad 5

Capa de Aplicación

SSH (Secure Shell)

- Permite ejecutar comandos en una máquina remota a través de una consola de comandos.
- Modelo cliente servidor (Ubuntu: openssh-server y openssh-client).
- TCP puerto 22.
- Manual Linux: “man ssh”
- Protocolo de intercambio de claves Diffie-Hellman y encriptación de clave pública

SCP (secure copy)

- Basado en SSH.
- Permite copiar archivos desde una máquina a otra.
- `scp usuario@IP_origen:/ruta /ruta_destino`
 - `scp usuario@IP_origen:/ruta/archivo /ruta_destino`
 - `scp /ruta/archivo usuario@IP_destino:/ruta_destino`

DNS (Domain Name System)

- Desventajas de usar direcciones IP para identificar máquinas servidores:
 - Recordar direcciones IP es complejo.
 - Si el servidor se mueve a otra máquina con otra IP, hay que avisar a todos los clientes.
- Se necesita un mecanismo para convertir nombres a direcciones IP.
 - Las máquinas no comprenden nombres.
- Primer antecedente: archivo **hosts.txt** (RFC 606).
 - Archivo disponible online por el Network Information Center (NIC).
 - Lista con nombres de computadoras (servidores) e IP correspondientes.
 - Los que deseaban ofrecer servicios debían anotarse.
 - Los usuarios debían descargar el archivo a sus computadoras.
 - Problemas:
 - Tamaño del archivo hosts.txt que todos debían descargar.
 - Conflictos por los nombres.
 - Cuello de botella y punto de falla.

DNS (Domain Name System)

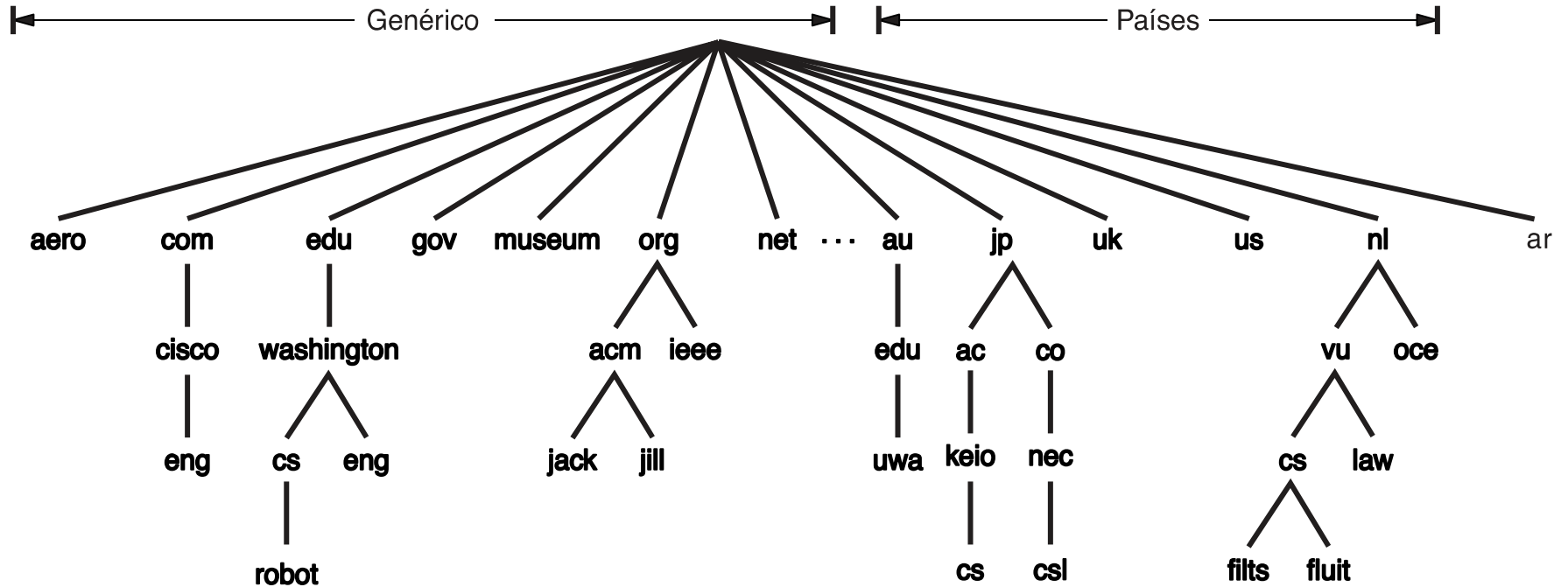
- RFC 1034 (inicial) + muchas otras.
- **Base de datos distribuida, jerárquico** basado en “**dominios**”.
- Aplicación cliente-servidor sobre **UDP**.
 - Cliente (resolvedor) envía una solicitud de DNS al servidor local indicando un nombre de dominio.
 - Procedimiento usualmente de biblioteca incluido en todos los SO o lenguajes. Primitivas como “gethostbyname”.
 - Puerto DNS: 53 (asignado por IANA).
 - El servidor busca la IP asociada al nombre de dominio y la informa al cliente.

DNS (Domain Name System)

- Administrador del **nivel superior** de la jerarquía de DNS: **ICANN**¹
- Nivel superior: 250 dominios, divididos en dos grupos:
 - **Por países** (ISO 3166)
 - **Genéricos**
 - Anteriores a 1980
 - Solicitados directamente a la ICANN.
- Los dominios pueden estar divididos en **subdominios de diversas jerarquías** (2, 3, 4, etc. jerarquías).
 - Nomenclatura: desde el más pequeño al más genérico.
 - **ingenieria.uncuyo.edu.ar**
 - Cada dominio controla los dominios debajo de él.

¹Internet Corporation for Assigned Names and Numbers

DNS (Domain Name System)



DNS (Domain Name System)

- Solo pueden utilizarse nombres disponibles y que no sean marcas comerciales registradas.
- Algunos dominios tienen restricciones (edu, mil, gov), otros no (com, net).
- Algunos problemas:
 - Ciberocupación.
 - Ejemplos de dns en venta en 2020: www.dinero.com.ar, www.2me2you.com, <http://www.badasscars.com>, <http://www.dinerodesdecasa.com/>
 -
 - Arriendo comercial.



DNS (Domain Name System)

	Uso	Restricciones	Ejemplo
.com	Comercial	No	google.com
.edu	Instituciones educativas	Si	uncuyo.edu.ar
.mil	Militar	Si	code.mil
.gov	Gobierno	Si	mendoza.gov.ar
.aero	Transporte aereo	Si	apcon.aero
.net	Proveedores de red, comercial	No	coches.net
.biz	Negocios	No	educacion.biz/
.org	Organizaciones sin fines de lucro	No	change.org
.xxx	adultos	No	

DNS (Domain Name System)

Registros de recursos de dominios

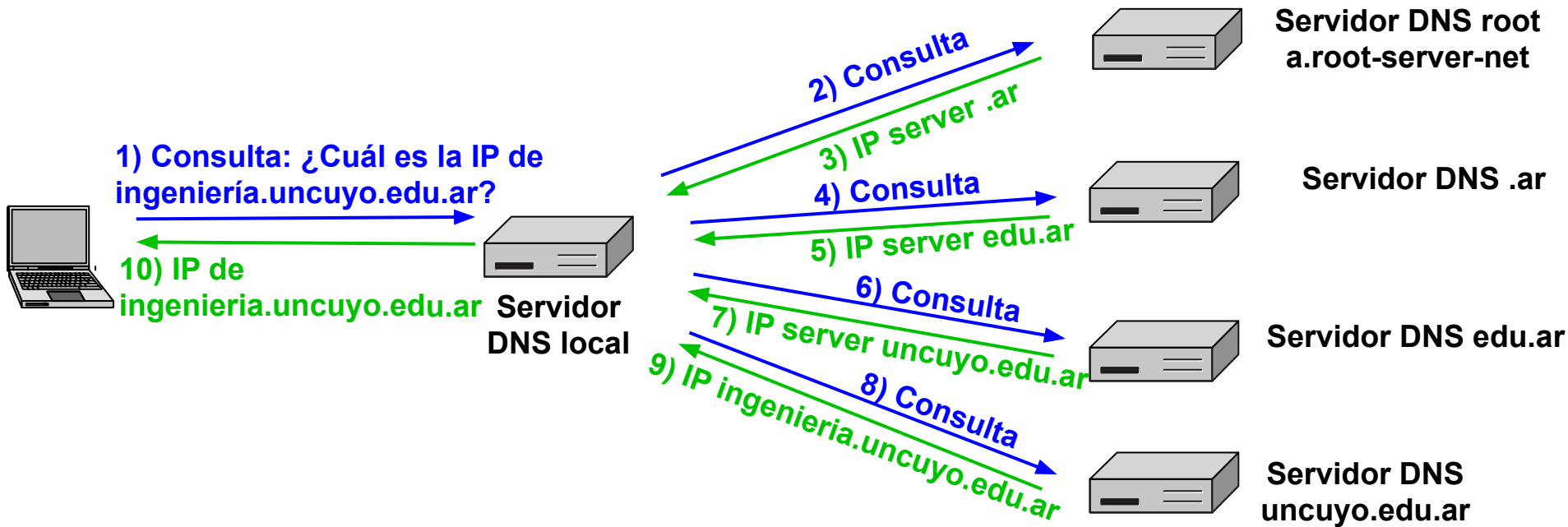
- Cada dominio posee un conjunto de **registros de recursos**.
- Los registros poseen la siguiente información:
 - Nombre de dominio
 - Tiempo de vida: En segundos. Indica la estabilidad del registro. Mientras más estable la información, mayor el valor.
 - Clase: Para Internet vale IN. Otros valores no se utilizan.
 - Tipo:
 - A: Dirección IPv4
 - AAAA: Dirección IPv6
 - NS: Nombre de un servidor para este dominio.
 - Cname: canonical name. El verdadero dns de la IP. DNS permite asociar varios “alias” a un mismo DNS, para ofrecer diferentes servicios.
 - Valor: Valor del registro.



Servidores de nombres

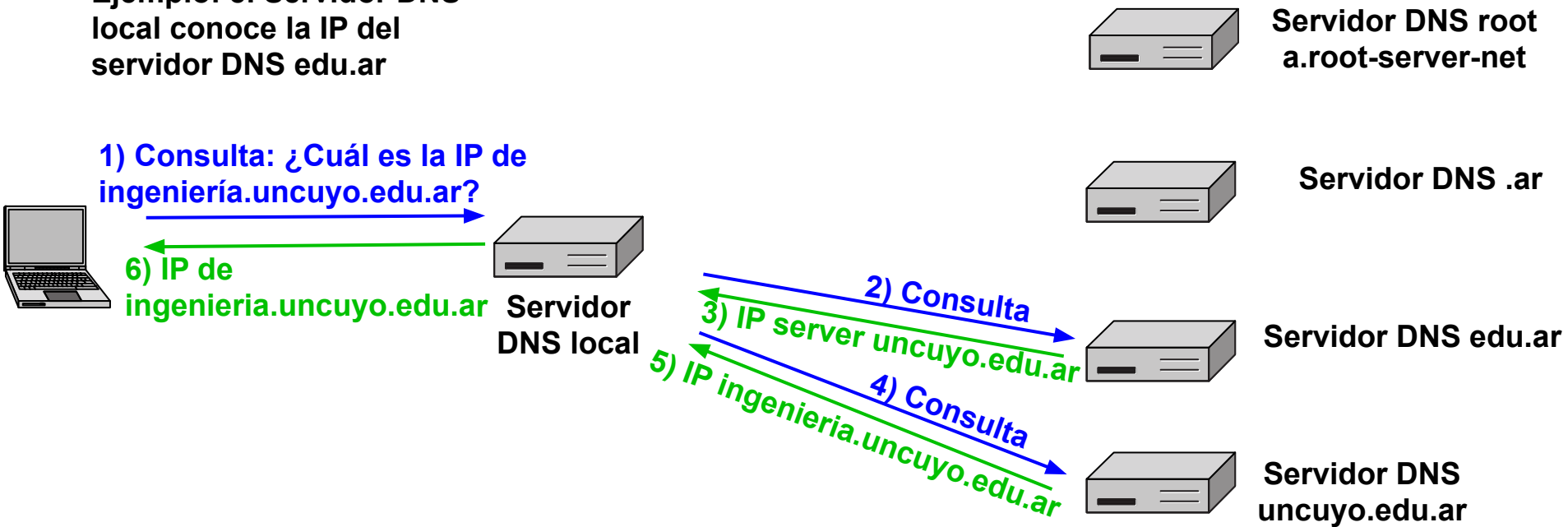
- Sistema distribuido y jerárquico.
- Nivel más alto: **13 servidores raiz** (<https://root-servers.org/>).
 - Nombres: **letra-root-servers.net**, donde letra va de a a m. (ejemplo: a-root-servers.net, b-root-servers.net,).
 - Cada servidor raíz es un sistema distribuido con servidores replicados en muchos países.
 - Direccionamiento anycast.
- Servidores de los dominios de mayor jerarquía:
 - Dominios genéricos
 - Dominios por países
- Servidores de menor jerarquía.
- El servidor DNS local es el responsable de encontrar a IP correspondiente a un nombre de dominio

DNS (Domain Name System) Proceso de resolución de nombres

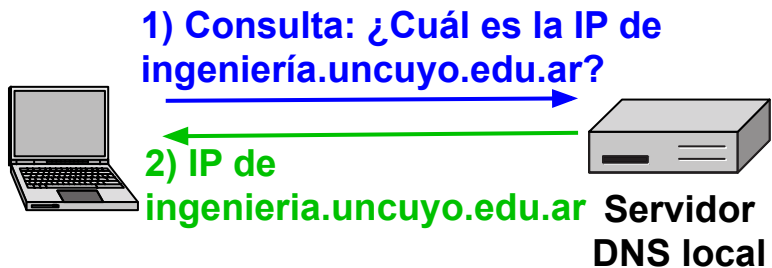


DNS (Domain Name System) Proceso de resolución de nombres + Caché

Ejemplo: el Servidor DNS local conoce la IP del servidor DNS edu.ar



DNS (Domain Name System) Proceso de resolución de nombres + Caché



Problema: ¿Que pasa si una asignación DNS-IP cambia?
Solución: Registros autorizados vs Registros en caché



Servidor DNS root
a.root-server-net



Servidor DNS .ar



Servidor DNS edu.ar



Servidor DNS
uncuyo.edu.ar

DNS (Domain Name System)

Proceso de resolución de nombres + Caché

Registros autorizados: Los brindados por la autoridad inmediata superior. Ejemplo: El registro autorizado de `ingenieria.uncuyo.edu.ar` es el que brinda el servidor `uncuyo.edu.ar`



Servidor DNS root
`a.root-server-net`



Servidor DNS .ar



Servidor DNS edu.ar



Servidor DNS
`uncu.edu.ar`

1) Consulta: ¿Cuál es la IP de `ingeniería.uncuyo.edu.ar`?



2) IP de

`ingeniería.uncuyo.edu.ar`



**Servidor
DNS local**

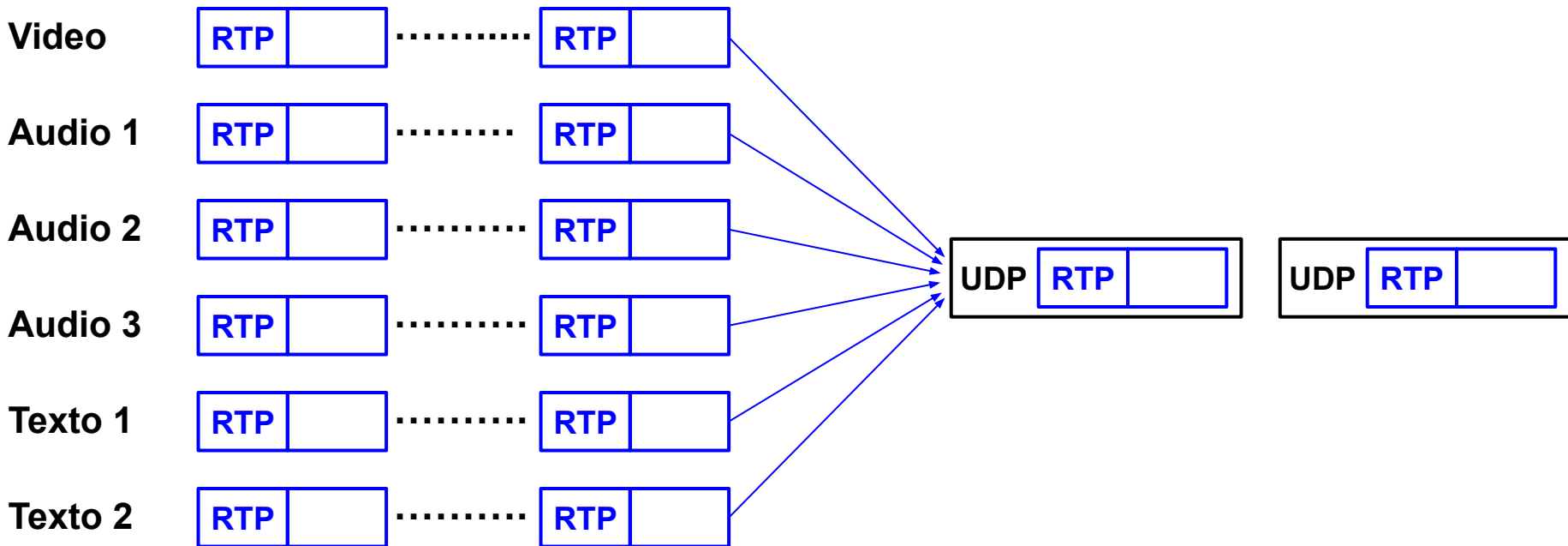


No es una autoridad de `ingeniería.uncuyo.edu.ar`, su registro será válido por el tiempo indicado en el campo tiempo de vida. Luego, el registro expirará, y una consulta a `ingeniería.uncuyo.edu.ar` comenzará nuevamente el proceso

RTP (Real-time Transport Protocol) y RTCP (Real Transport Control Protocol)

- Multimedia en tiempo real (RFC 3550, año 2003).
 - Telefonía y videoconferencias en tiempo real.
 - Radio sobre Internet.
 - Música y video bajo demanda.
- Transporta datos de audio y video en paquetes.
- Mayoritariamente UDP.
- Una aplicación multimedia consiste en varios flujos de datos:
 - Uno de video
 - Varios de sonido:
 - Dos o más para sonido estéreo o envolvente.
 - Sonido en varios idiomas.
 - Texto (subtítulos)

RTP (Real-time Transport Protocol) y RTCP (Real Transport Control Protocol)

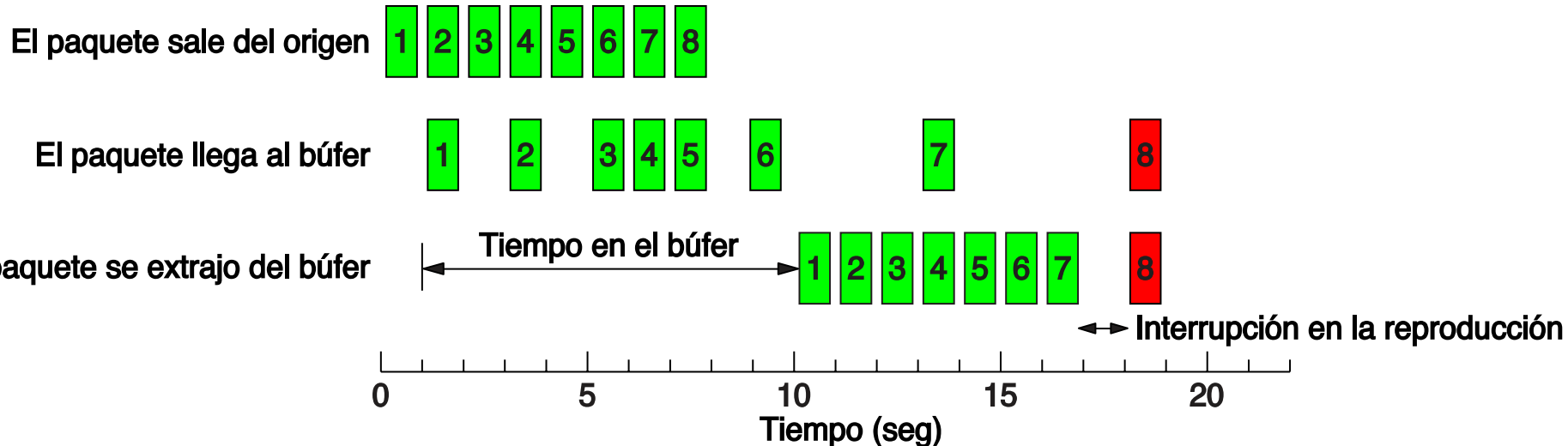


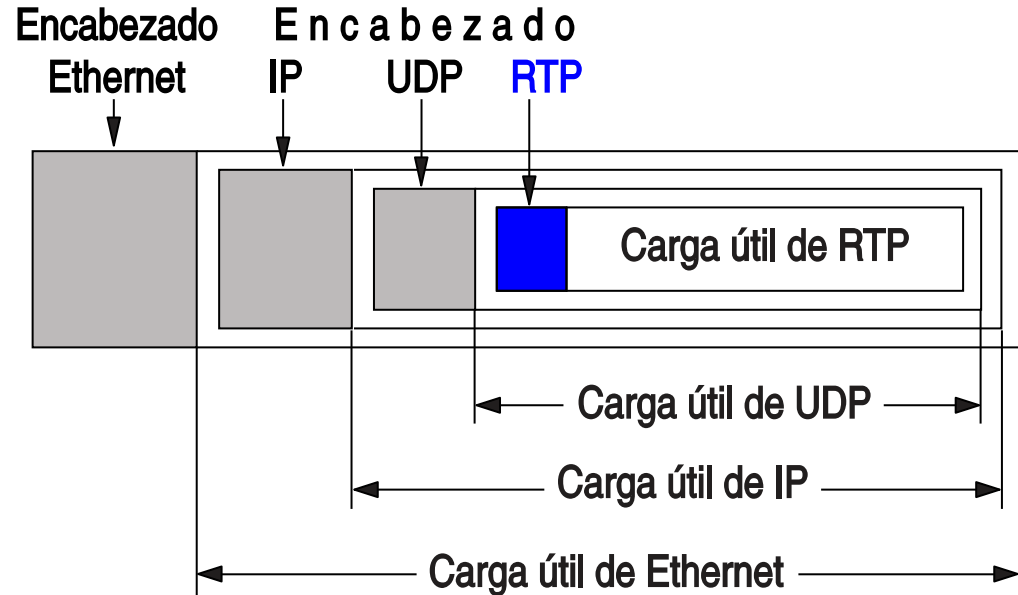
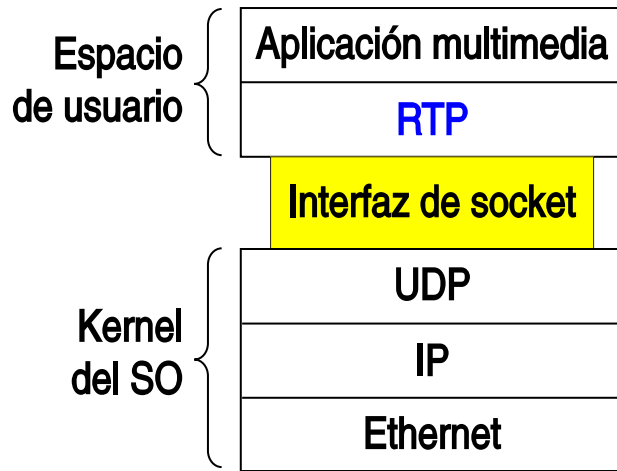
RTP y RTCP

- Cada paquete posee un **número de secuencia** y una **estampa de tiempo**.
 - Número de secuencia: permite ordenarlos.
 - Si un paquete se pierde, puede ignorarse o deducirse su valor desde otros paquetes (para audio), nunca retransmitirse.
 - Estampa de tiempo:
 - Permite que cada paquete se reproduzca en el momento justo.
 - Permite sincronizar diferentes flujos.
 - Son relativas al primer paquete.
- Cada flujo puede codificarse en diferentes formas (MP3, MP4, PCM, etc.), y el código puede cambiar dinámicamente según el ancho de banda disponible en la red (por eso los paquetes indican el código utilizado en el campo carga útil).

RTP y RTCP

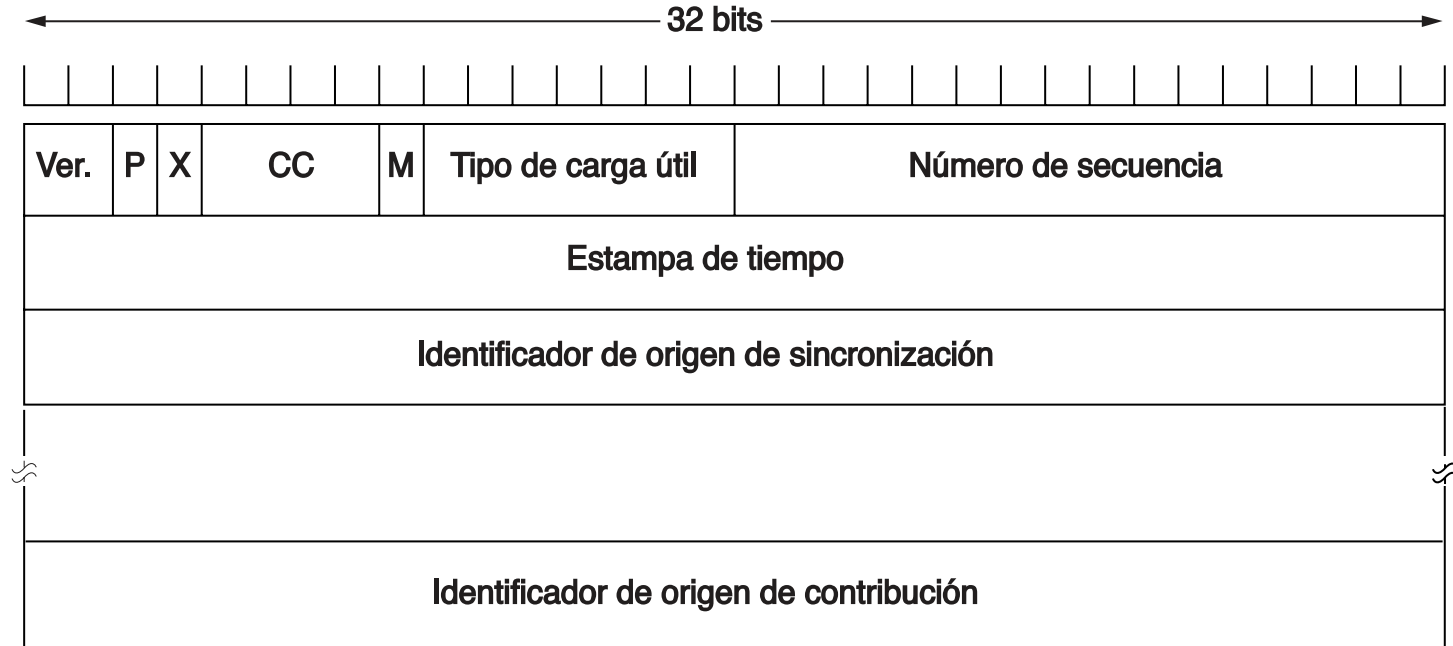
- **Jitter**: Variación en el retardo. Parámetro crítico en multimedia.
- Solución: **buffer en el receptor** que almacena los paquetes y permiten retrasar la reproducción.
- Tiempo en el buffer: parámetro crítico que depende de cada aplicación.







RTP (Real-time Transport Protocol) y RTCP (Real Transport Control Protocol)

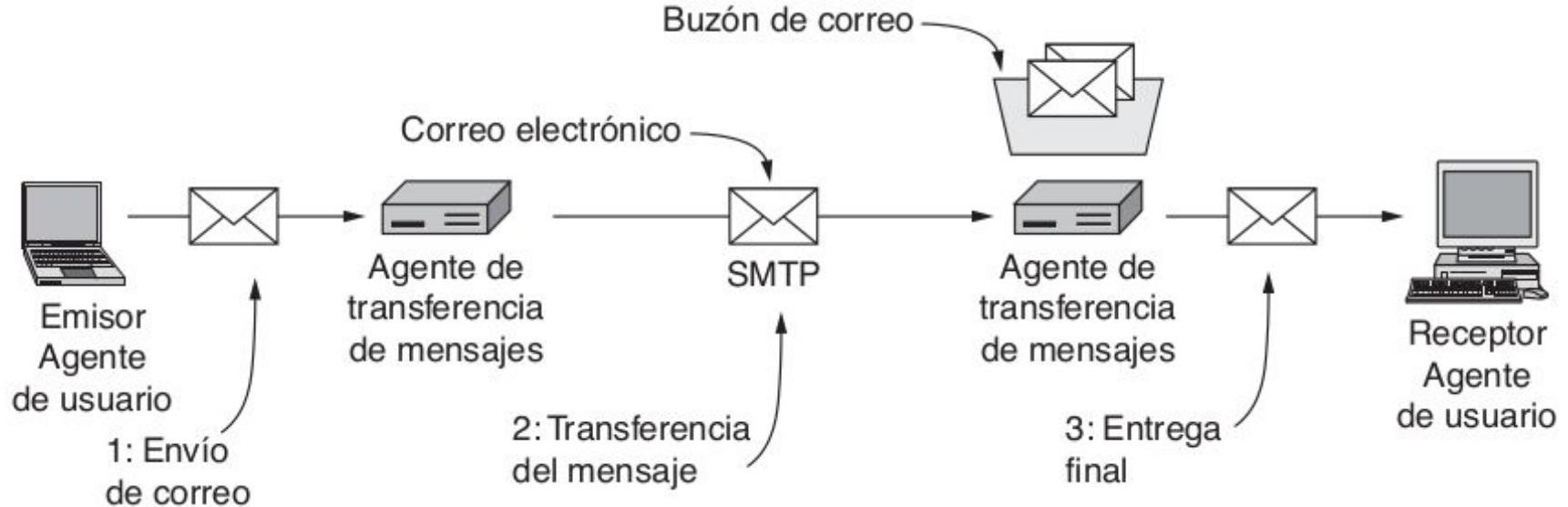


RTP y RTCP

- Ver.: Versión
- P: Indica si se han agregado bytes de relleno (utilizado para cifrado).
- X: Hay encabezado de extensión.
- CC: Número de fuentes de contribución.
- M: Bit que se deja para que la aplicación lo utilice.
- Tipo de carga útil: Algoritmo de codificación utilizado para los datos de la carga útil (RFC 1890).
- Origen de sincronización: Puede haber un origen diferente para la sincronización.
- Origen de contenidos: Lista de orígenes del contenido.
- RTCP: Envía mensajes de retroalimentación al emisor indicando que:
 - Puede aumentar o debe disminuir la velocidad de transmisión
 - El emisor puede cambiar el algoritmo de codificación.

SMTP

- SMTP (Simple Mail Transfer Protocol, Protocolo Simple de Transferencia de Correo). RFC 821 y RFC 5321.





SMTP

- Se encarga de:
 - Mover el e-mail desde el origen al destino.
 - Notificación de estado de la entrega y errores.
- Basado en el sistema DNS (registros MX): usuario@dirección-dns.
- Trabaja sobre TCP.
- No realiza cifrado ni autenticación.
- Mejoras.
 - Versión original: Solo texto ASCII de 7 bits.
 - MIME (Extensiones Multipropósito de Correo Internet). Indica el contenido del e-mail, para que el cliente sepa si podrá mostrar el contenido, o necesitará instalar plugins o ejecutar aplicaciones externas.
 - ESMTP: Extensiones a SMTP (RFC 1869).
- Otros protocolos que intervienen:
 - IMAP (actual) o Pop3 (anterior): Protocolo para entrega de correos a la aplicación cliente final (aplicación de usuario).

WWW (World Wide Web)

- Enorme **colección de contenido** en forma de documentos (archivos) llamadas páginas web. Estos archivos contienen:
 - Vínculos a otras páginas web del mundo
 - Instrucciones de como presentar el contenido
 - Instrucciones de emplear lenguajes encriptados.
- **ES un gran sistema distribuido** en millones de máquinas.
- Comenzó en el CERN (Centro europeo para la investigación nuclear) en 1989.
 - Tim Berners-Lee
- Consorcio W3C (World Wide Web Consortium):
 - Creado en 1994 por el CERN y el MIT.
 - Crear protocolos.
 - Fomentar la interoperabilidad.
 - www.w3.org

WWW (World Wide Web)

- Los documentos se escriben usando **lenguaje HTML** (HyperText Markup Language o lenguaje de marcas de hipertexto).
- Sigue el modelo **cliente-servidor**:
 - El cliente (navegador) solicita al servidor una página web (archivo html).
 - El servidor envía la página web.
 - El cliente la interpreta la página web.
- El protocolo que controla el proceso de intercambio de documentos web se denomina **HTTP (HyperText Transfer Protocol)**.
 - Trabaja sobre **TCP puerto 80 (http) o 443 (https)**.

WWW (World Wide Web)

- **Navegador web:** Intérprete HTML y lenguajes encapsulados (scripts).
 - Generar texto, imágenes, formatos, etc., de acuerdo a lo indicado en la página mediante HTML.
 - Pedir contenido a otros sitios.
 - Ejecutar secuencias de comandos (lenguajes incrustados).
- Páginas web pueden ser
 - **Estáticas:** son siempre iguales.
 - **Dinámicas:** No son siempre iguales.
 - Un programa en el servidor las genera cuando el cliente las solicita.
 - Ejecuta instrucciones en el servidor o el cliente que modifican la página.

Primer navegador gráfico: Mosaic, luego mejorado y llamado Netscape Navigator.
Google Chrome y Firefox: botón derecho y luego “ver código fuente”

WWW (World Wide Web)

- **URL (Uniform Resource Locator):** identifica de manera unívoca como llegar a un recurso (página web, imagen, video, etc.)
- Tres partes:
 - Protocolo (http, https, ftp, mailto, file, etc, about:.)
 - DNS: www.ingenieria.uncuyo.edu.ar.
 - Ruta al recurso (index.html).
 - Ejemplo: <http://www.ingenieria.uncuyo.edu.ar/index.php>
 - **Desventaja:** Los servidores web actuales son distribuidos y poseen caché, por lo que sería útil referenciar la página web sin indicar la ubicación.
- **URI (Uniform Resource Identifiers):**
 - URI que indican cómo llegar al recurso: URL
 - URI que no indican cómo llegar al recurso: URN (Uniform Resource Names)
 - Ejemplo URN: Doi (Digital Object Identifier) <https://www.doi.org/>
 - 10.1016/S0169-7552(98)00110-X
 - Reglas para escribir URI: RFC 3986.

WWW (World Wide Web) Extensiones

- Los tipos de archivos que los proveedores desean proveer a los clientes son cada vez más variados (video, imágenes, etc.).
- Muchos tipos de documentos pueden ser manejados por los navegadores, pero algunos no. Soluciones:
 - Complementos (plug-ins, extensiones, etc.).
 - Llamar a aplicaciones auxiliares.
- Cuando un servidor devuelve una página web, **también devuelve información sobre el tipo de contenido** (llamados tipos MIME¹).

¹ Multipurpose Internet Mail Extensions (Extensiones Multipropósito de Correo Internet). Fueron creadas para el correo electrónico, luego utilizadas para páginas web. Ejemplos: text/html, image/jpeg, image/png, audio/mpeg, video/mp4, application/xml, application/pdf

WWW (World Wide Web)

¿Que hace el navegador cuando escribimos la URL?

- Verifica si hay cookies asociados con el DNS solicitado.
- Obtiene la IP del servidor mediante DNS.
- Solicita una conexión TCP al puerto 80 (HTTP) o 443 (HTTPS) de la máquina indicada.
- Realiza una petición HTTP incluyendo la ruta a página web indicada.
 - Anexa información adicional (navegador y versión, idioma, etc.).
- Obtiene el archivo con el código html y lo interpreta.
- Obtiene las URLs de recursos incluidos en la página (imágenes, etc.) y realiza las correspondientes peticiones HTTP.
- Si la conexión TCP permanece inactiva, la libera.
- Ejecuta secuencias de instrucciones en algún lenguaje incrustado (Javascript).
 - Pueden ser ejecutadas al inicio o por un evento ejecutado por el usuario (apretar un botón, etc.).
- Permite ejecutar muchos protocolos sobre el navegador (ftp, video, etc.)

WWW (World Wide Web)

Servidores HTTP

- Espera y acepta conexiones TCP al puerto 80 (http) o 443 (https).
- Recibe peticiones HTTP y
 - Obtiene la página solicitada (archivo escrito con código HTML) e información adicional (navegador del usuario, idioma del usuario, cookies, etc).
 - Página por defecto: index.html o index.php
 - Ejecuta secuencias de instrucciones en algún lenguaje incrustado (PHP ASP, CGI) para generar o modificar la página web.
- Determina información adicional a incluir en la respuesta (tipos MIME).
- Envía la respuesta empleando http o https.
 - Puede haber diferentes respuestas según navegador del usuario, idioma, dispositivo utilizado (PC, teléfono móvil, etc).
- Servidores web populares: Apache.



HTTP

- Consisten en una o más líneas de texto ASCII.
- GET: Solicita un recurso (por ejemplo, una página web). Puede incluir datos de entrada a ser procesados por el servidor en la URL.
 - GET nombreadarchivo HTTP/1.1
 - <https://www.google.com/search?q=palabra>
 - <http://ingenieria.uncuyo.edu.ar/buscar/index?terminos=calendario>
- POST: Envía datos al servidor indicando el archivo que almacena el código que debe procesar esos datos. Los datos se envían en el cuerpo de la solicitud (usualmente los almacena, realiza algún cálculo, etc.).
- HEAD: Solicita solo el encabezado de la respuesta (para conocer de antemano los datos a recibir y características de los mismos).



```
250 17.917324785 192.168.0.105 179.0.132.51 TCP 66 47384 → 80 [ACK] Seq=1 Ack=1 Win=64
251 17.917710261 192.168.0.105 179.0.132.51 HTTP 1072 GET /index.php HTTP/1.1
252 17.921609689 179.0.132.51 192.168.0.105 TCP 74 80 → 47386 [SYN, ACK] Seq=0 Ack=1 W

▶ Frame 251: 1072 bytes on wire (8576 bits), 1072 bytes captured (8576 bits) on interface wlo1, id 0
▶ Ethernet II, Src: LiteonTe_59:47:93 (24:fd:52:59:47:93), Dst: Tp-LinkT_a6:8b:34 (ac:84:c6:a6:8b:34)
▶ Internet Protocol Version 4, Src: 192.168.0.105, Dst: 179.0.132.51
▶ Transmission Control Protocol, Src Port: 47384, Dst Port: 80, Seq: 1, Ack: 1, Len: 1006
▼ Hypertext Transfer Protocol
  ▼ GET /index.php HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): GET /index.php HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /index.php
      Request Version: HTTP/1.1
      Host: fcen.uncuyo.edu.ar\r\n
      Connection: keep-alive\r\n
      Upgrade-Insecure-Requests: 1\r\n
      User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari
      Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
      Accept-Encoding: gzip, deflate\r\n
      Accept-Language: es-419,es;q=0.9\r\n
    ▼ [truncated]Cookie: _gid=GA1.3.396630582.1619985333; _ga_KG6NHTCE0W=GS1.1.1621229589.61.1.1621231822.0; _ga=
      Cookie pair: _gid=GA1.3.396630582.1619985333
      Cookie pair: _ga_KG6NHTCE0W=GS1.1.1621229589.61.1.1621231822.0
```




323	18.363625479	192.168.0.105	179.0.132.51	TCP	66 47384 → 80 [ACK] Seq=1007 Ack=18144
324	18.363634170	179.0.132.51	192.168.0.105	HTTP	499 HTTP/1.0 200 OK (text/html)
325	18.364114389	192.168.0.105	179.0.132.51	TCP	66 47384 → 80 [FIN, ACK] Seq=1007 Ack=1
326	18.368450710	192.168.0.105	179.0.132.51	TCP	71 47384 → 80 [SYN] Seq=0 Win=64240 Len=

```

Frame 324: 499 bytes on wire (3992 bits), 499 bytes captured (3992 bits) on interface wlo1, id 0
Ethernet II, Src: Tp-LinkT_a6:8b:34 (ac:84:c6:a6:8b:34), Dst: LiteonTe_59:47:93 (24:fd:52:59:47:93)
Internet Protocol Version 4, Src: 179.0.132.51, Dst: 192.168.0.105
Transmission Control Protocol, Src Port: 80, Dst Port: 47384, Seq: 18144, Ack: 1007, Len: 433
[15 Reassembled TCP Segments (18576 bytes): #292(1428), #294(1428), #296(581), #298(1428), #300(1428), #302(1428)]
Hypertext Transfer Protocol

```

```

  HTTP/1.0 200 OK\r\n
    Date: Mon, 17 May 2021 07:12:11 GMT\r\n
    Server: Apache/2.2.16 (Debian)\r\n
    X-Powered-By: PHP/5.3.3-7+squeeze19\r\n
    [truncated]Set-Cookie: icb=a%3A5%3A%7Bs%3A10%3A%22session_id%22%3Bs%3A32%3A%221e3aa6e5dfb220473d4ad7be60aa8a\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
  Content-Length: 17890\r\n
    [Content length: 17890]
    Content-Type: text/html\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.445923909 seconds]
    [Request in frame: 251]
    [Request URI: http://fcen.uncuyo.edu.ar/index.php]
    Content-encoded entity body (gzip): 17890 bytes -> 114352 bytes
    File Data: 114352 bytes

```

Line-based text data: text/html (1840 lines)

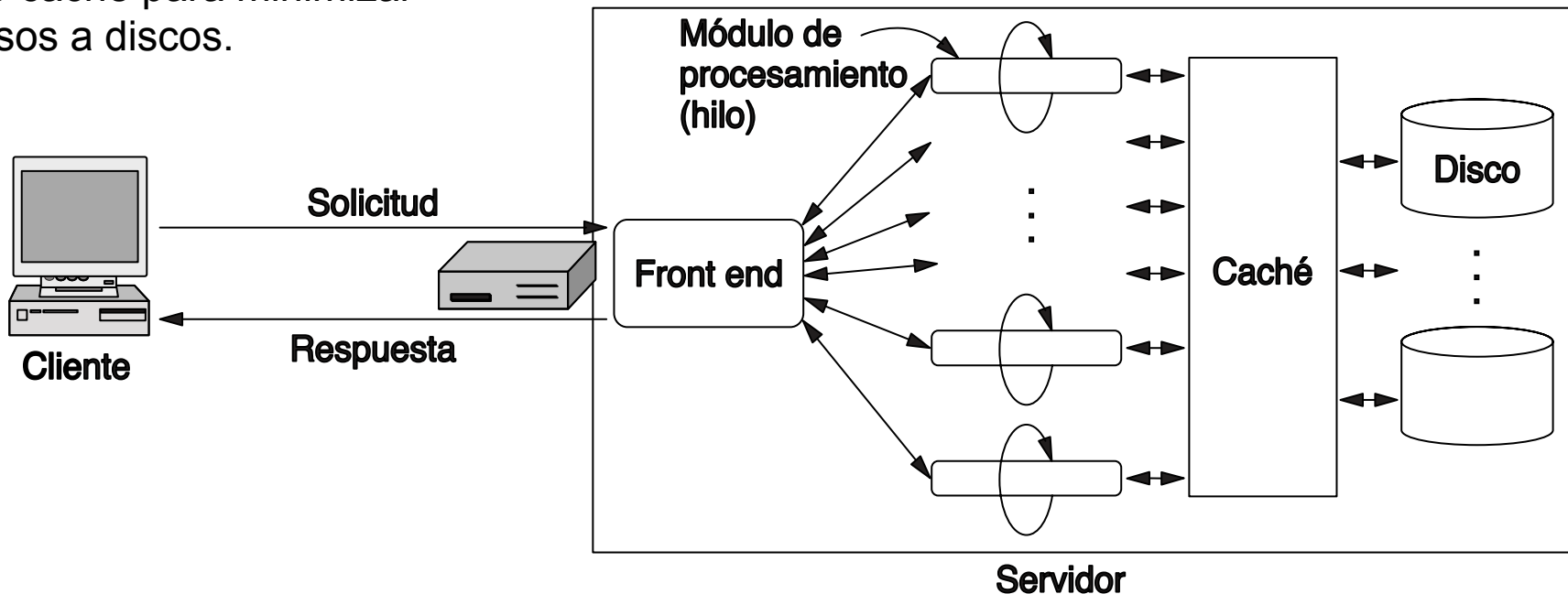
```

<!DOCTYPE html>\n
<html lang="es">\n
<head>\n
<title> ESTILOS Y JAVASCRIPT COMUNES </title>\n

```

Necesidad de gran cantidad de memoria RAM que actuará como caché para minimizar accesos a discos.

WWW (World Wide Web) Servidores multihilos



WWW (World Wide Web) HTML (HyperText Markup Language)

Estructura:

- Basado en marcas o etiquetas.
 - `esto va en negrita`
- Permite el ingreso de datos
 - Formularios
- Permite enviar datos al servidor para que los procese algún programa
 - POST
 - GET

```
<html>
<head> </head>
<body>

<form ACTION="url programa" method=POST>
<input type=text name="cliente">
<input type=submit value="nombre">
</form>
</body>
</html>
```



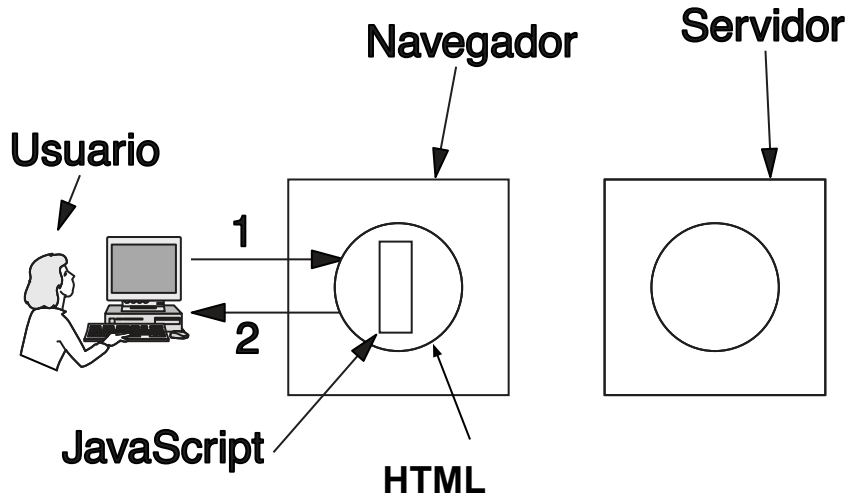
Evolución de HTML

Elemento	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0	HTML 5.0
Hipervínculos	x	x	x	x	x
Imágenes	x	x	x	x	x
Listas	x	x	x	x	x
Mapas e imágenes activas		x	x	x	x
Formularios		x	x	x	x
Ecuaciones			x	x	x
Barras de herramientas			x	x	x
Tablas			x	x	x
Características de accesibilidad				x	x
Incrustación de objetos				x	x
Hojas de estilo				x	x
Secuencias de comandos				x	x
Video y audio					x
Gráficos vectoriales en línea					x
Representación de XML					x
Hilos en segundo plano					x
Almacenamiento del navegador					x
Lienzo de dibujo					x

Páginas Web dinámicas

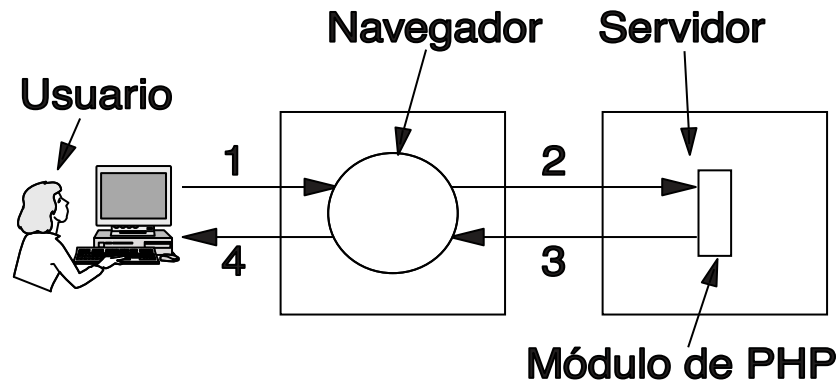
Ejecución de instrucciones del lado del cliente:

JavaScript, VBScript, applets



Ejecución de instrucciones del lado del servidor

PHP, ASP, CGI



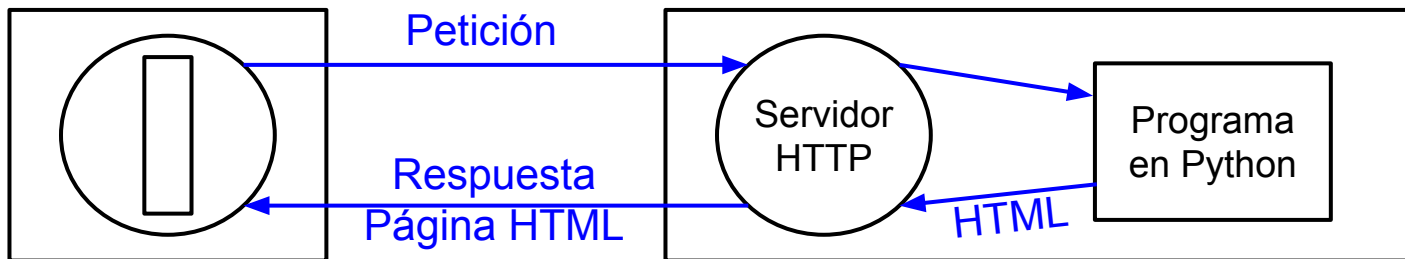
Ejecución en el cliente: Ejemplo programa JavaScript

```
<html>
<head>
<script language="JavaScript" type="text/javascript">
function respuesta(mi_form) {
var mensaje=document.getElementById("nombre_id").value;
alert("usted escribio: " + mensaje);
}
</script>
</head>
<body>

<form name="mi_form" id="form_id" method="POST" onsubmit="return respuesta(this.form);">
<input type="text" name="nombre" value="" id="nombre_id">
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

Ejecución del lado del Servidor: CGI

- CGI (Common Gateway Interface o Interfaz de Puerta de Enlace Común).
 - RFC 3875.
 - Interfaz que permite a un servidor web comunicarse con programas externos que pueden aceptar entradas y generar páginas HTML en respuesta.
 - Pueden estar escritos en cualquier lenguaje (Python, Ruby, Perl, C++, etc.)



Ejecución del lado del Servidor: PHP

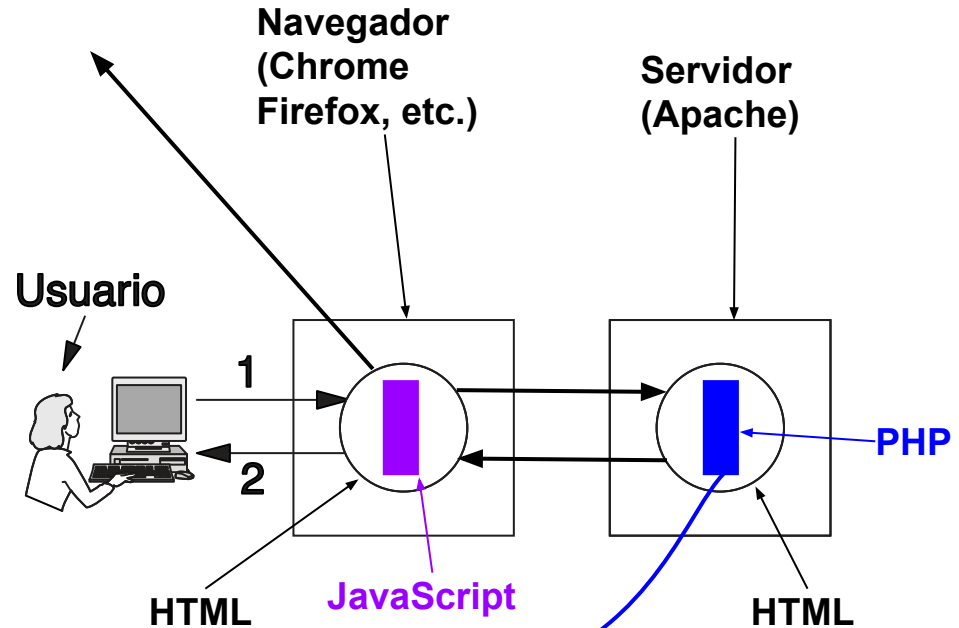
- PHP (Hypertext Preprocessor - Preprocesador de Hipertexto).
- Permite incrustar secuencias de instrucciones php dentro de documentos HTML, que el servidor ejecuta.
 - El servidor debe interpretar PHP.
- Puede elegir entre varias secuencias de código HTML para generar la respuesta, o imprimir datos, etc.

- Otras opciones a PHP o CGI: JSP (JavaServer Pages), ASP (versión Microsoft).
- Frameworks:
 - Lado cliente (HTML, Javascript, SSC): Apache Cordova, React .js (Facebook), Angular (Google), etc.
 - Lado servidor: Symfony (PHP), .NET (ASP), etc.



```
<form name="form1" ... action="verificacion.php" method="post" onsubmit="return control_datos();">
  <input type="text" name="usuario" id="usuario_id">
  <input type="password" name="clave"><br><br>
  <input type="submit">
</form>
```

```
<body>
  <?php
    $clave_correcta="abcde";
    $usuario = $_POST["usuario"];
    $clave = $_POST["clave"];
    if($clave==$clave_correcta){
      .....
    }
    else{
      .....
    }
  ?>
```



XML y JSON

- Especifican **contenido** estructurado (HTML especifica formato+contenido).
- No hay etiquetas definidas. Cada usuario puede definir sus propias etiquetas.
- Interpretados de manera Estricta.
- **XML** (eXtensible Markup Language)
 - Utiliza etiquetas (similar a HTML).
 - Pensado como estándar para comunicación entre aplicaciones (dentro y fuera de WWW).
- **JSON** (JavaScript Object Notation)
 - Utiliza notación similar a Javascript ({};"" etc.)
 - Creado para intercambio de datos JavaScript, pero aceptado por muchos languages.
- **AJAX** (Asynchronous JavaScript And XML): XML+JavaScript (o JSON + JavaScript). Permite intercambiar información en segundo plano entre el servidor y el cliente en formato XML y mostrarlo dinámicamente mediante JavaScript (sin actualizar la página completa).



Ejemplo XML

```
<?xml version="1.0" ?>
  <lista_libros>
    <libro>
      <titulo> Human Behavior and </title>
      <autor> George Zipf </autor>
      <anio> 1949 </anio>
    </libro>
    <libro>
      <titulo> The Mathematical Theory </title>
      <autor> Claude E. Shannon </autor>
      <anio> 1949 </anio>
    </libro>
    <libro>
      <titulo> Nineteen Eighty-Four </title>
      <autor> George Orwell </autor>
      <anio> 1949 </anio>
    </libro>
  </lista_libros>
```

HTML: Hojas de estilo en cascada (CSS: Cascading Style Sheets)

- Inicialmente HTML tenía el objetivo de mostrar información, sin permitir controlar la apariencia.
- Cuando se hizo importante la apariencia, se agregaron más etiquetas.
 - Ejemplos `` (negrita), `` (tipo de letra), `<i>` (cursiva o itálica), etc.
 - El código se hizo tedioso de escribir o leer y no portable¹.
- Hojas de estilo: Asocian “texto - estilo lógico”. Los estilos lógicos se definen por separado y son opcionales (el navegador puede utilizarlas o no).
 - Ejemplo: “texto - encabezado segundo orden”.
 - Los estilos pueden agregarse dentro del mismo documento html (etiqueta `<style>`) o en archivos diferentes de tipo css.

¹ Una página diseñada para un navegador puede hacerse no apta para otro navegador o pantalla.

HTML: Hojas de estilo en cascada (CSS: Cascading Style Sheets)

- Ejemplos de definiciones css:

```
body {background-color:linen; color:navy; font-family:Arial;}
```

```
h1 {font-size:200%}
```

```
h2 {font-size:150%}
```

- Ejemplo de inclusión:

```
<head>
```

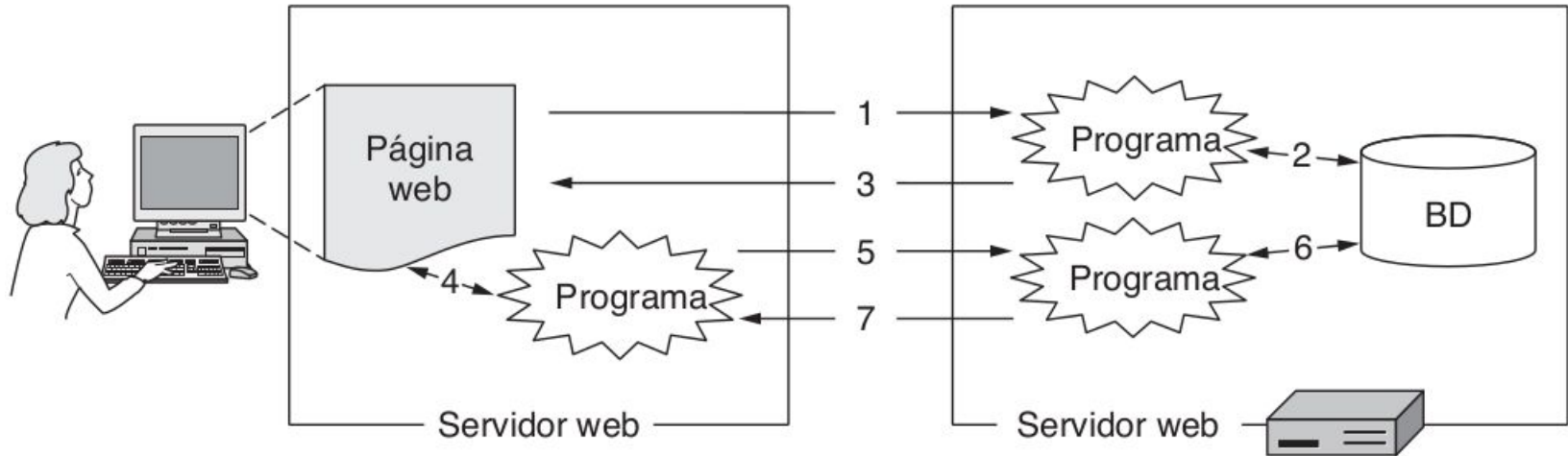
```
<title>WIDGETS AMALGAMADOS, S.A. </title>
```

```
<link rel="stylesheet" type="text/css" href="awistyle.css" />
```

```
</head>
```

Aplicaciones Web (Computación en la nube)

- Aplicaciones que se ejecutan parte en el servidor y parte en el navegador web del usuario.
 - El navegador web puede ejecutar la interfaz de usuario.
 - Rivalizan o sustituyen a aplicaciones tradicionales.



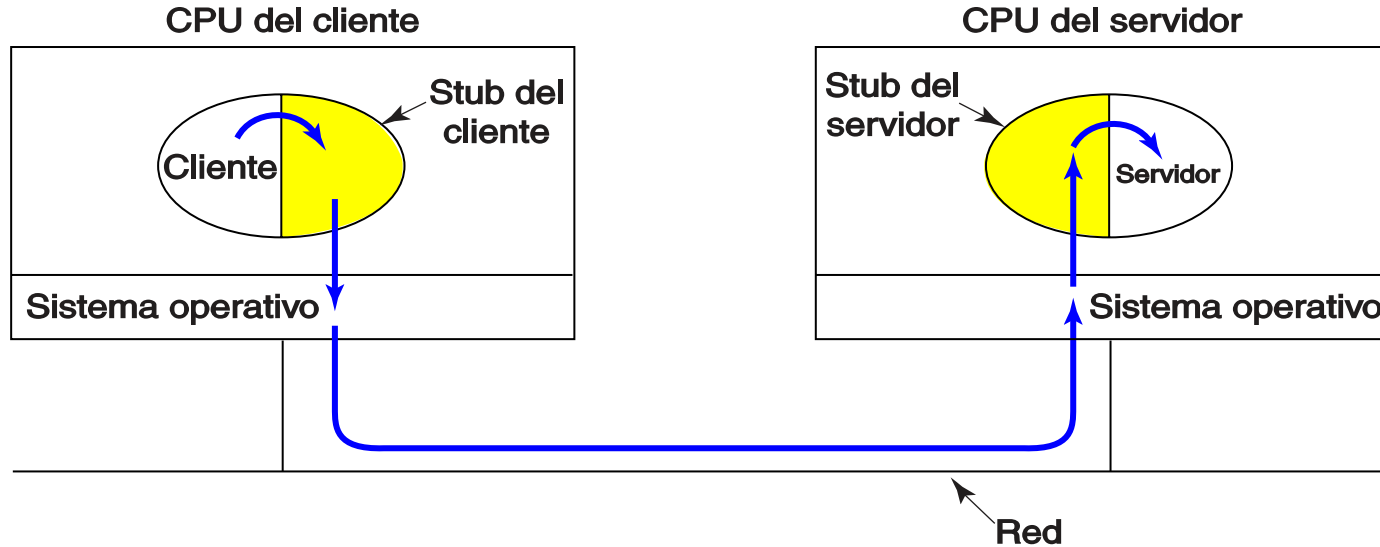
RPC (Remote Procedure Call)

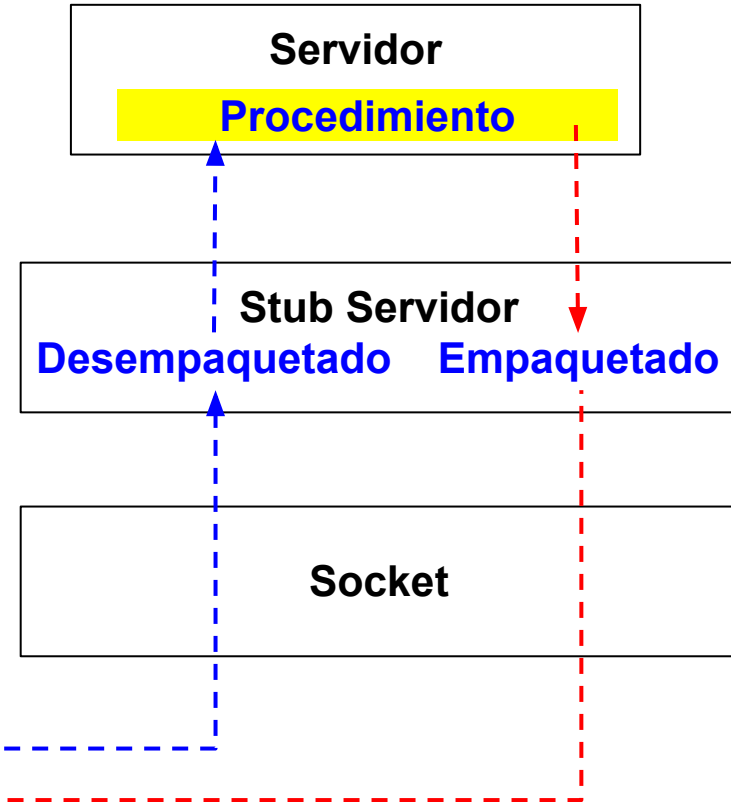
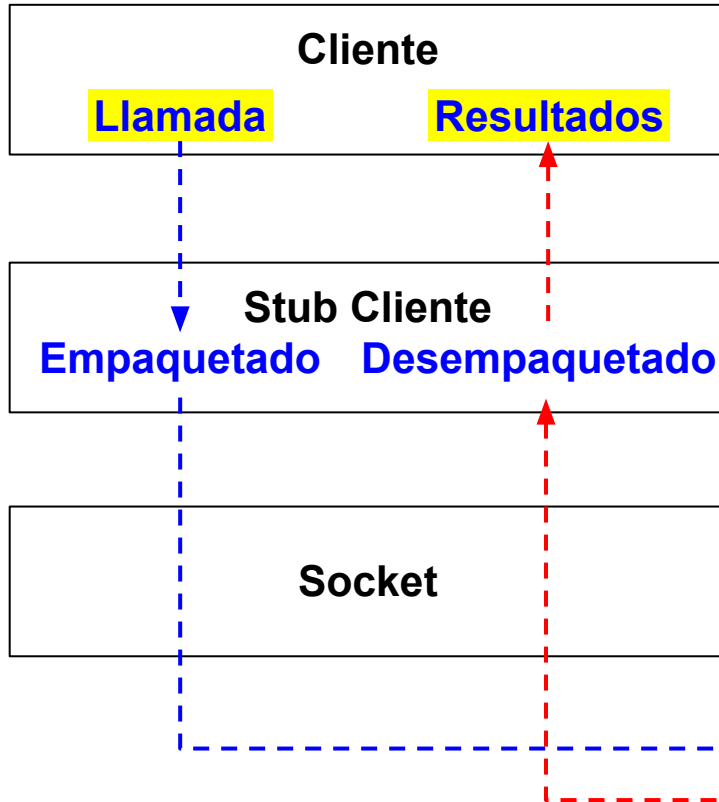
- Un proceso **llamado cliente** puede llamar procedimientos que **están en otra máquina como si fueran locales**.
- El proceso que contienen los procedimientos remotos se llama **servidor**.
- La llamada puede incluir el paso de datos como **argumentos** y el procedimiento remoto puede **devolver resultados** (como una llamada a un procedimiento local).
- La llamada, los parámetros y los resultados deben **encapsularse en paquetes** que deben enviarse al servidor. Este proceso debe ser transparente.
- Emplea mayormente **UDP**, ya que la llamada, argumento y respuestas implican pocos datos, **pero pueden perderse paquetes y fallar la llamada**.
- También puede emplear **TCP**, en casos donde la llamada no puede fallar:
 - No todas las funciones pueden llamarse dos veces (cada llamada puede modificar dato, como un contador).

RPC (Remote Procedure Call)

- Necesidad de un procedimiento **stub de cliente** que representa o simula al procedimiento que está el servidor:
 - Permite llamar al procedimiento remoto como si fuera local.
 - Empaqueta la llamada y los datos pasados como parámetros en un paquete que envía a la máquina remota.
 - Implementa los mecanismos para transmitir el paquete.
- Necesidad de un procedimiento **stub de servidor**.
 - Realiza la llamada al procedimiento como si fuera local (aunque la llamada fue en otra máquina).
 - Empaqueta los resultados de la ejecución del procedimiento y los envía a al proceso cliente.
- **Desventajas:**
 - **No se pueden utilizar punteros (apuntadores).**

RPC







RPC Ejemplo Cliente usando Google Cloud Platform

```
registro_actividad.recuperar_registro(usuario, new AsyncCallback<String>(){
    @Override
    public void onFailure(Throwable caught) {
        actividad_TextArea.setText("ERROR recuperando datos");
    }
    @Override
    public void onSuccess(String result) {
        actividad_TextArea.setText("\nActividad realizada por el usuario " + usuario + ":\n" +
        result);
    }
}
```

RPC Ejemplo Servidor

(el procedimiento recuperar_registro se encuentra dentro de la clase registro_actividad)

```
public String recuperar_registro(String usuario) {  
    String answer = "";  
  
    .....  
    final Query query = new Query(Registro_actividad1.Registro_actividad1_ENTITY);  
    query.addSort(Registro_actividad1.DATE, SortDirection.ASCENDING);  
    query.addFilter(Registro_actividad1.USER, Query.FilterOperator.EQUAL, usuario);  
    .....  
    for (Entity entity : datastoreService.prepare(query).asIterable()) {  
        .....  
        for (Registro_actividad1 registro : registros_de_actividad_List) {  
            answer = answer + "\n" + registro.get_date() + " " + registro.get_user() .....  
        }  
    }  
    return answer;  
}
```

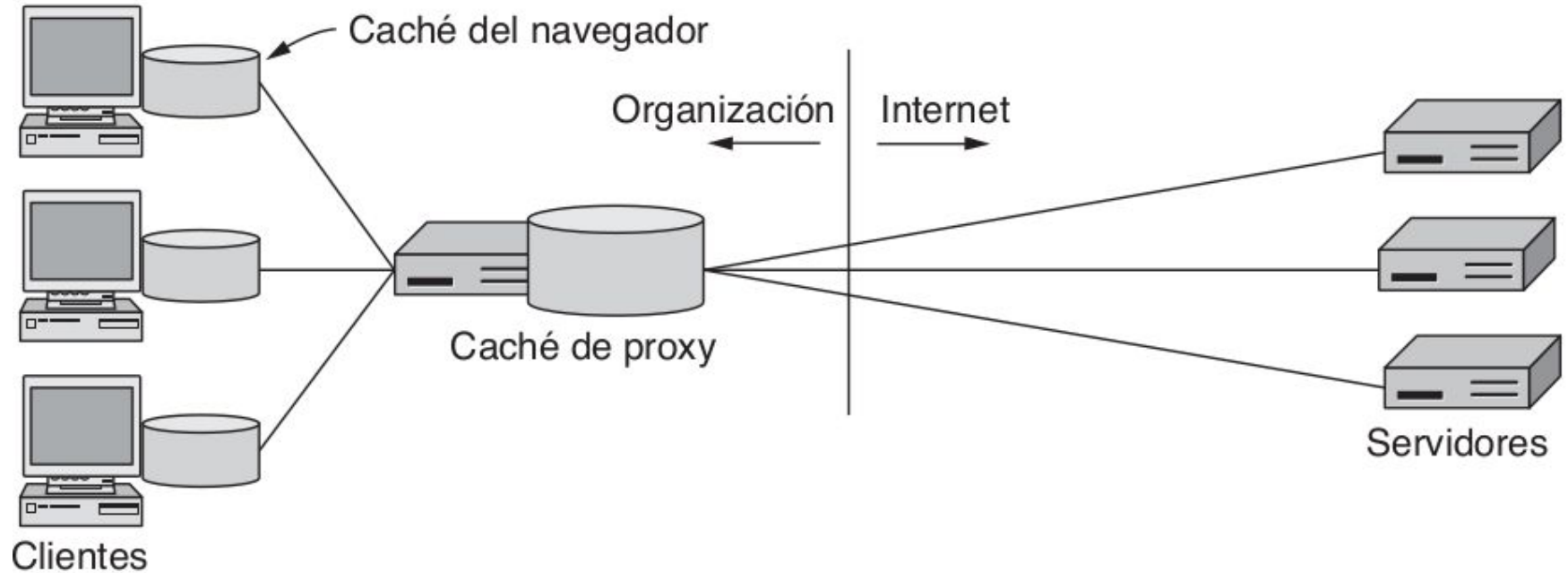
Caché Web

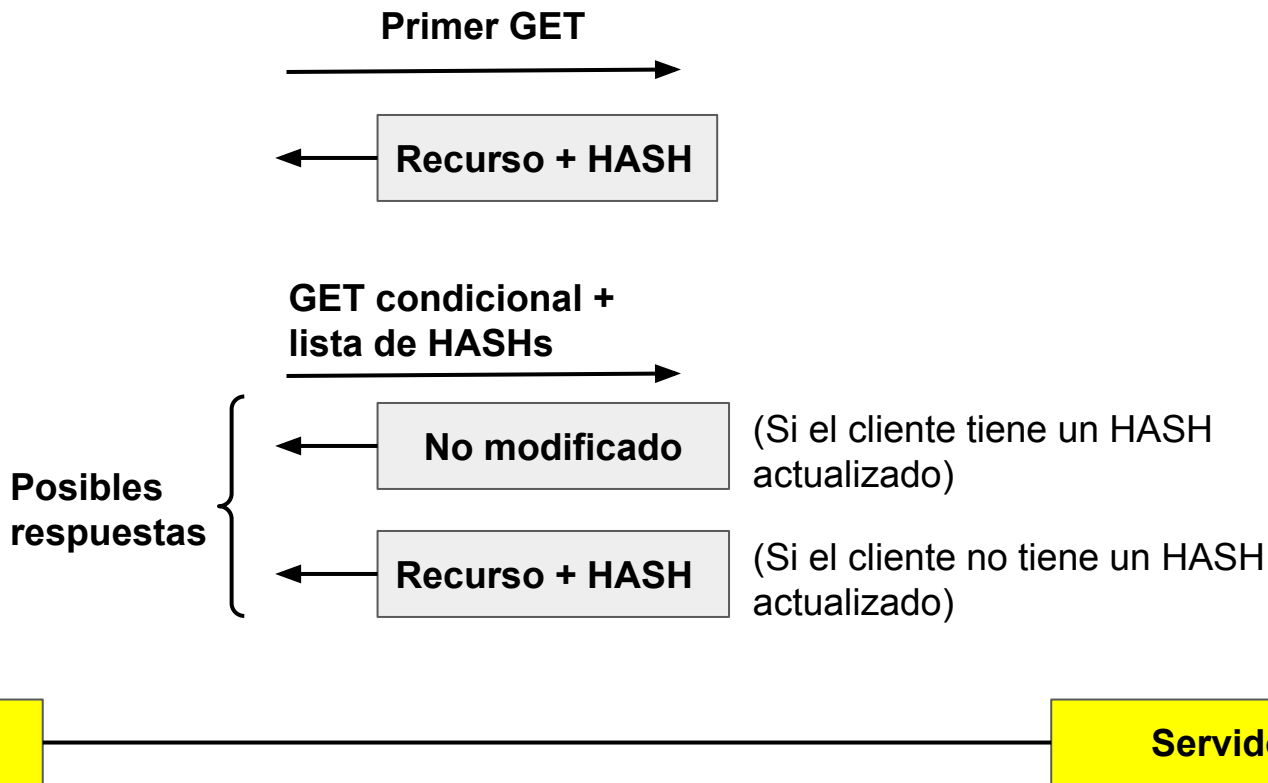
- **Objetivo:** Almacenar información de URL visitados (páginas web, imágenes, etc.) de manera que posibles futuros accesos sean más veloces.
 - Concepto similar a memoria caché del procesador.
- **La información puede almacenarse en:**
 - La computadora del usuario (caché del navegador).
 - Computadoras intermedias (caché proxy).
 - Usado por ISPs o empresas.
- **Problema de actualización de los datos (¿El dato en la caché fue o no actualizado en el servidor?).** Dos estrategias:
 - El servidor provee como dato la fecha y hora de expiración. Una segunda petición revisa si la URL existe en la caché y si sigue siendo válida. Si lo es, la recupera de la caché.
 - **Problema:** No todas las páginas incluyen datos sobre expiración.

Caché Web

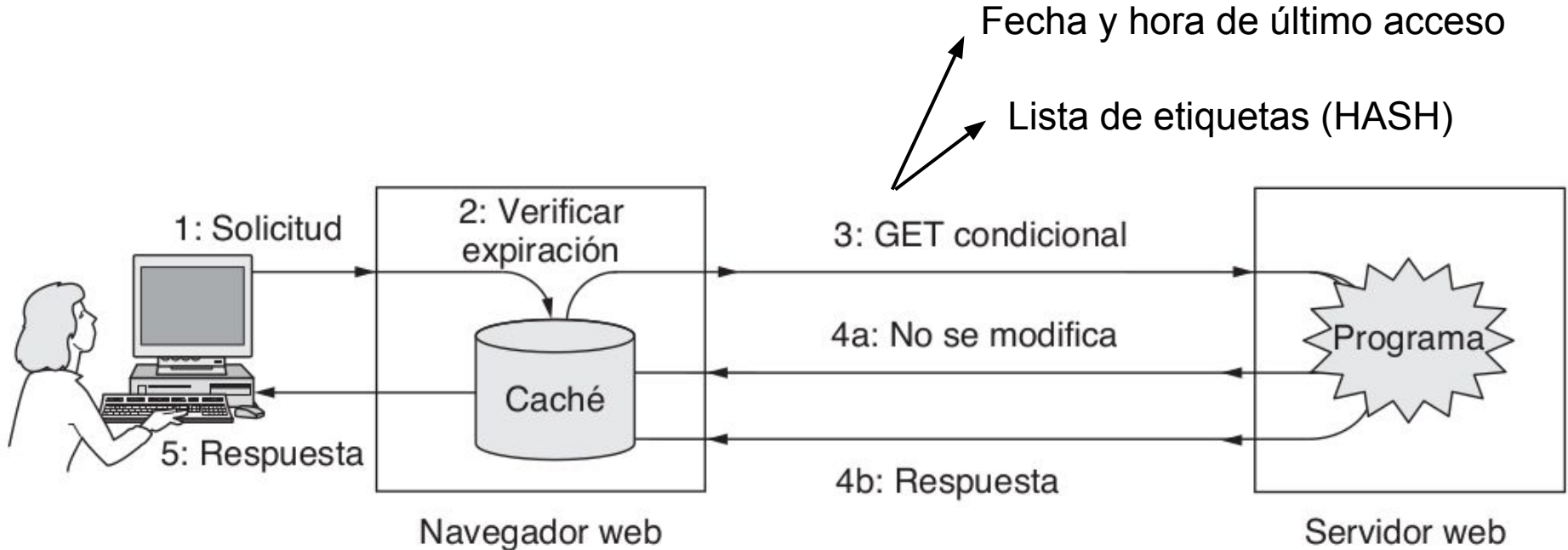
- GET condicional: Dos formas:
 - El cliente envía un GET indicando fecha y hora de última actualización.
 - Con cada envío de un recurso, el servidor envía una etiqueta de validez (hash llamados ETag). Luego, en el GET condicional, el cliente envía las etiquetas que posee, con las cuales el servidor puede determinar si el cliente posee una copia actualizada.
 - Si no hubo actualizaciones, el servidor lo indica, y el navegador toma la URL desde la caché.
 - Si hubo actualizaciones, el servidor envía el nuevo archivo.
- El servidor puede enviar directivas adicionales (ejemplo: no-caché).
- En un proxy web, puede haber varios niveles de caché (en una empresa, luego en el ISP).

Caché Web





Caché Web

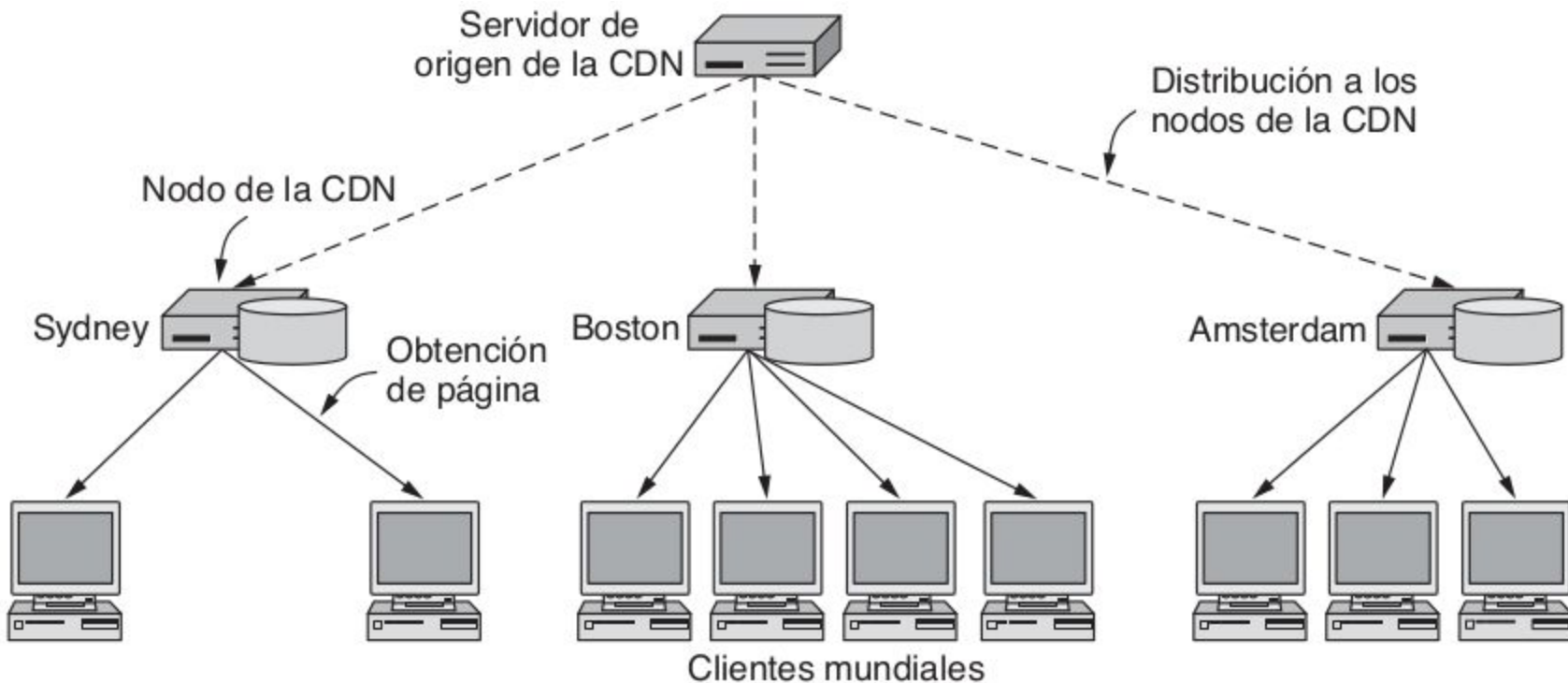


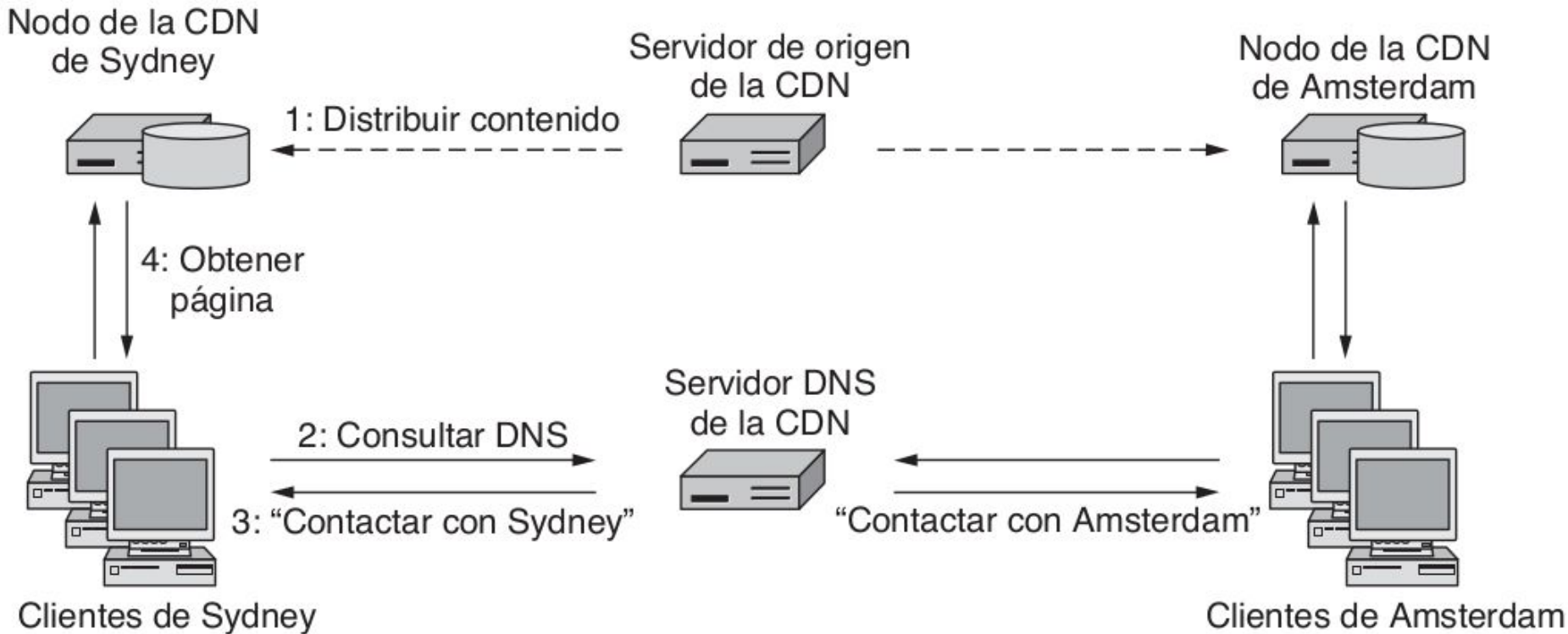
CDN (Content Delivery Networks)

- El proveedor de un servicio coloca copias de su contenido (página web) en nodos en distintas ubicaciones, e indica a los clientes usar esos nodos.
 - Concepto inverso a los proxy web.
- Estructura árbol.
- ¿Cómo obtiene el cliente la IP del nodo al que debe acceder?
 - Espejos: El servidor replica su contenido en los nodos, a los que se les llama espejos. La página web principal posee vínculos a los diferentes nodos.
 - Utilizado para distribuir grandes paquetes de software o contenido de gran tamaño.
 - Que el dueño de la CDN administre el servidor DNS. Devuelve la IP del nodo en función de:
 - La IP origen de la solicitud.
 - La carga de cada nodo.

CDN (Content Delivery Networks)

- Existen empresas que proveen CDN como servicio. El cliente provee el contenido, el propietario de la CDN lo distribuye entre sus nodos.
- Ventajas adicionales:
 - Mayor vulnerabilidad a picos de tráfico elevado.
 - No hay un servidor o red cuello de botella.





Redes P2P (Peer-to-Peer o Igual a Igual)

- **Objetivo:** Distribución de contenido.
- **Primer antecedente:** Napster (1999).
- Formadas por nodos (llamados pares o iguales) que actúan como cliente (para obtener contenido) y como servidores (para proveer contenido).
 - Todos los nodos actúan de igual manera para distribuir el contenido.
 - No hay una estructura centralizada.

Protocolo BitTorrent

- Los archivos a compartir se dividen en trozos de igual tamaño. El nombre de cada trozo se identifica con un hash (SHA-1 de 160 bits del trozo).
- **Trackers (rastreadores):** Para cada contenido, contiene una lista de nodos con trozos del contenido (hashes).
 - A medida que los clientes descargan trozos del contenido, informan al tracker que poseen el trozo, y pueden actuar como servidores para otros clientes.



Redes P2P

- Cada proveedor de contenido crea un archivo de descripción de contenido llamado “torrent” que contiene:
 - El nombre de un rastreador.
 - La lista de trozos contenido (hashes).
 - Se contactan con el rastreador periódicamente para enviarle esta información.
 - Sembradores: nodos que contienen todos los trozos de un contenido.
- Pasos para la descarga de un archivo:
 1. Obtener el torrent (de una página web o buscador del cliente).
 2. Localizar el rastreador y la lista de nodos con trozos del contenido.
 3. Descargar trozos y convertirse en proveedor de los trozos descargados.
 4. Cada nodo descarga diferentes trozos para que no haya nodos con pocas copias en la red.
- La velocidad de descarga será función de los trozos provistos a otros clientes.

Redes P2P

