

Introducción a Internet y a la Web

Clientes, servidores y la nube



◆ **Clientes:**

- Dan acceso a información y servicios en Internet

◆ **Servidores:**

- Alojan la información y los servicios

◆ **La nube:** conjunto de terminales y servidores

- interconectados con aplicaciones y protocolos de Internet

◆ **TCP/IP:** protocolos de interconexión de redes de fibra, cable, WIFI, ...

- sobre los que se implementan las aplicaciones de Internet y sus protocolos

Cientes, navegadores y tiendas



◆ **Cientes** de acceso a Internet más importantes

- PCs, portátiles, tabletas, teléfonos inteligentes

◆ **Navegador (browser)** cliente Web de acceso a servidores

- Utilizando: **URL, HTTP, HTML, CSS y JS**
 - ◆ p.e. Chrome, Firefox, Internet Explorer, Opera, Safari, ...



◆ **Tiendas de aplicaciones**

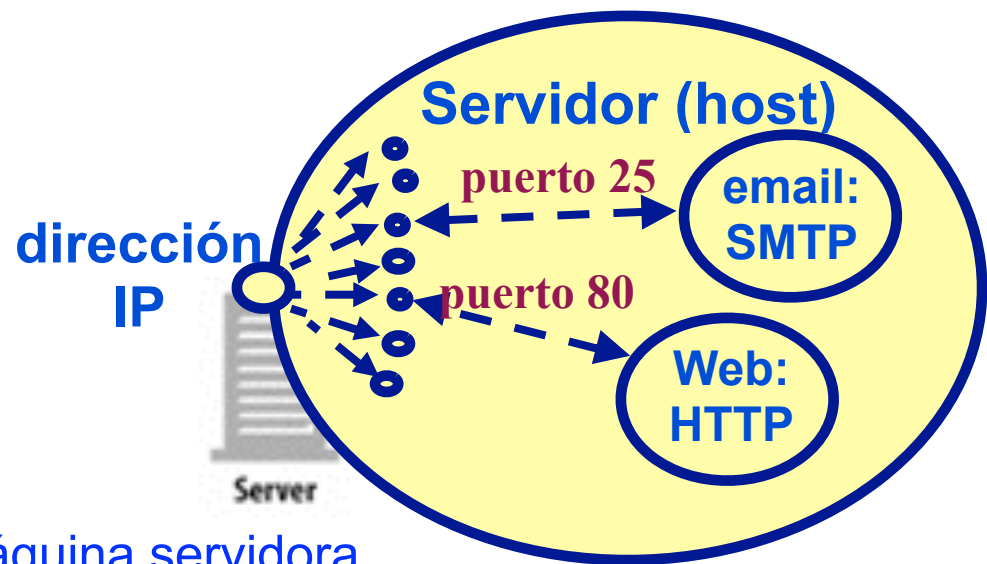
- Instalan aplicaciones en móviles y tabletas
 - ◆ Las aplicaciones usan las normas de la Web (URL, HTTP,)

Máquina servidora (host)



- ◆ Contiene información y servicios
- ◆ Una máquina servidora tiene una dirección “**conocida**” en Internet
 - Dirección simbólica (de dominio o **DNS**): **upm.es**, **google.com**, ...
 - ◆ Cada dirección de dominio tiene una dirección IP (binaria) asociada
- ◆ Hay 2 tipos de direcciones IP: IPv4 e IPv6
 - **IPv4**: versión 4 del protocolo IP con dirección de **32 bits** o **4 octetos**
 - ◆ ejemplo: **192.9.0.144**, **127.0.0.1** (localhost - mi máquina), ...
 - **IPv6**: versión 6 del protocolo IP (última) con dirección de **128 bits**
 - ◆ ejemplo: **2001:db8:85a3::8a2e:370:7334**,

Servidores y puertos



◆ Puerto

- Dirección de 16 bits dentro de la máquina servidora
 - ◆ Es donde se instala el **programa servidor**
- El **programa servidor** es lo que normalmente denominamos **servidor**
 - ◆ Cliente y servidor se comunican a través de un protocolo: HTTP, SMTP,
 - Utilizando el interfaz de sockets TCP/IP para comunicar entre ambos

◆ Los servicios tienen un protocolo y un puerto por defecto

- **Web:** protocolo HTTP (puerto 80), HTTPS (443)
- **Email:** protocolo SMTP (puerto 25), POP3 (110), IMAP143)
- **Shell segura:** protocolo SSH (puerto 22)

◆ Si un servidor no está en el puerto por defecto

- Su dirección debe incluir el puerto, p.e. dit.upm.es:8080, 192.9.0.144:8080

URL



◆ URL (Uniform Resource Locator)

- Dirección de un recurso en un servidor en Internet
- ◆ Internet soporta muchos tipos de servicios diferentes
 - Cada tipo de servicio utiliza un URL y protocolo diferentes
- ◆ Algunos ejemplos de tipos de URLs
 - **URL Web:** utiliza HTTP para acceder a recursos, incluye
 - ◆ Protocolo, servidor y recurso (camino): <http://google.com/picture.png>
 - **URL de correo (email):** identifica el buzón de usuario, incluye
 - ◆ protocolo, buzón de usuario y servidor: mailto:pepe_garcia@gmail.com

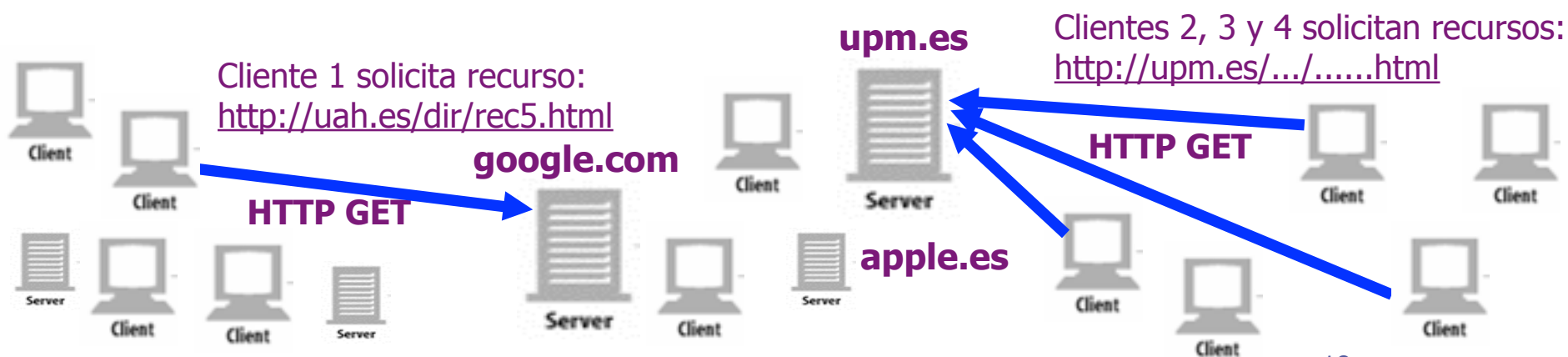
HTTP (HiperText Transfer Protocol)

◆ Protocolo del Web

- Procesa recursos identificados por un URL en un servidor remoto

◆ Métodos o comandos principales de HTTP

- **GET:** trae al cliente (lee) un recurso identificado por un **URL**
- **POST:** crea un recurso identificado por un URL
- **PUT:** actualiza un recurso identificado por un URL
- **DELETE:** borra un recurso identificado por un URL
- (hay mas comandos)



Aplicación Web

- ◆ Aplicaciones ejecutables en un navegador creadas con
 - **HTML, CSS y JavaScript**



- ◆ **HTML**

- Lenguaje de marcado de páginas Web
 - ◆ define la estructura del contenido de una página Web
- En WebApps define la interfaz de la aplicación con el usuario

- ◆ **CSS**

- Define el estilo visual de un una página o aplicación Web (HTML)

- ◆ **JavaScript**

- Lenguaje de programación de aplicaciones de cliente

Aplicación Web: HTML, CSS y JavaScript

◆ HTML

- Lenguaje de marcado

◆ CSS

- Estilo la visualización

◆ JavaScript

- Lenguaje de programación

```
02-date_CSS.htm UNREGISTERED
<!DOCTYPE html><html>
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8">
  <style type="text/css">
    body {color: blue;}
  </style>
</head>
<body>
  <h3>Fecha y hora</h3>
  <script type="text/javascript">
    document.write(new Date());
  </script>
</body>
</html>
```



WebApps o aplicaciones de cliente

◆ Aplicaciones que **residen en un servidor**

- pero se **ejecutan en un cliente**

- ◆ Se identifican con un **URL**: `http://upm.es/apps/webapp.html`

- El cliente trae la aplicación del servidor con el **protocolo HTTP (GET)**

◆ Las apps se construyen con las tecnologías de la Web

- URLs, HTTP, HTML, CSS y JavaScript

1) Cliente solicita WebApp identificada con URL

2) Script se ejecuta al cargar la página Web en el navegador:



Solicitud HTTP GET asociada a un URL

Respuesta HTTP: página Web con script



Servidor sirve fichero identificado por URL

```
<!DOCTYPE html><html>
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8">
  <style type="text/css">
    body {color: blue;}
  </style>
</head>
<body>
  <h3>Fecha y hora</h3>
  <script type="text/javascript">
    document.write(new Date());
  </script>
</body>
</html>
```





¿Para qué se utiliza el **CSS**?

- a) Es el lenguaje de programación de aplicaciones de cliente.
- b) Define la estructura de la información de una página web.
- c) Define el estilo de visualización de una página Web en el navegador.



Para cada una de las siguientes afirmaciones, indique si es **verdadera** o **falsa**



- a) **GET** es un comando para traer páginas Web a un cliente para su visualización.
- b) **POST** es un comando para editar recursos en un servidor.
- c) **PUT** es un comando para crear recursos en un servidor.
- d) **DELETE** es un comando para borrar recursos en un servidor.



Test de autoevaluación.

Para cada una de las siguientes afirmaciones, indique si es **verdadera** o **falsa**



- a) **HTTP** permite solicitar recursos a un servidor web.
- b) Una **URL** identifica sin ambigüedad un recurso en Internet.
- c) El código **JavaScript** que define la lógica de las aplicaciones cliente no puede ser embebido en las páginas HTML.
- d) **HTML** define la estructura de la información de una página web.