



2º SMR  
***Aplicaciones Web***

# **UD1: Introducción a la Web**

---

**IES Segundo de Chomón**



### ÍNDICE

1. Aplicaciones web
  - 1.1. Internet
  - 1.2. La web
  - 1.3. Página web
  - 1.4. Sitio web
  - 1.5. Aplicación web
2. Evolución histórica de la web
3. Navegadores y servidores web



### APLICACIONES WEB

#### Aplicación Web

- Herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de **internet** mediante un **navegador**
- **Programas** que funcionan en **internet**
  - No necesitan ser instalados en el ordenador
- Las aplicaciones web se apoyan en **Internet** y la **Web**



Blogs, wikis y foros.



Redes sociales.



Netflix:



### 1.1. INTERNET

#### Internet

- **Internet** es un conjunto descentralizado de **redes** de comunicación **interconectadas** que utilizan la familia de protocolos TCP/IP
- Red de redes
  - Funciona como una **red lógica única**, de alcance **mundial**, aunque esté formada por multitud de redes físicas heterogéneas.
- Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como **ARPANET**, entre tres universidades en California y una en Utah, Estados Unidos

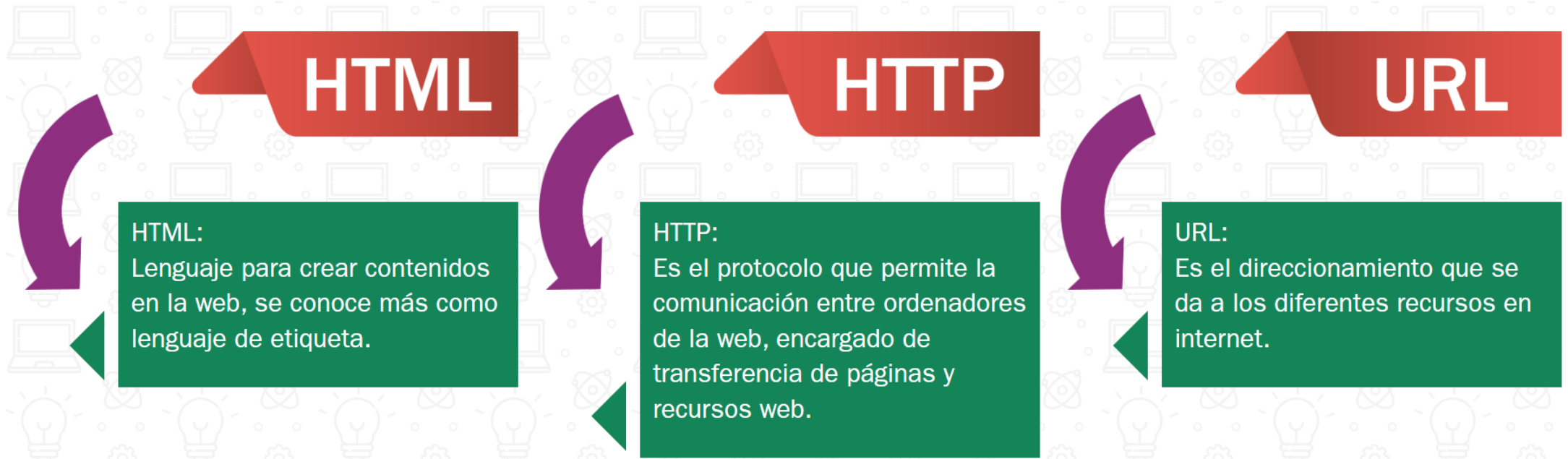




### LA WEB

#### ➤ World Wide Web, WWW, o la Web”

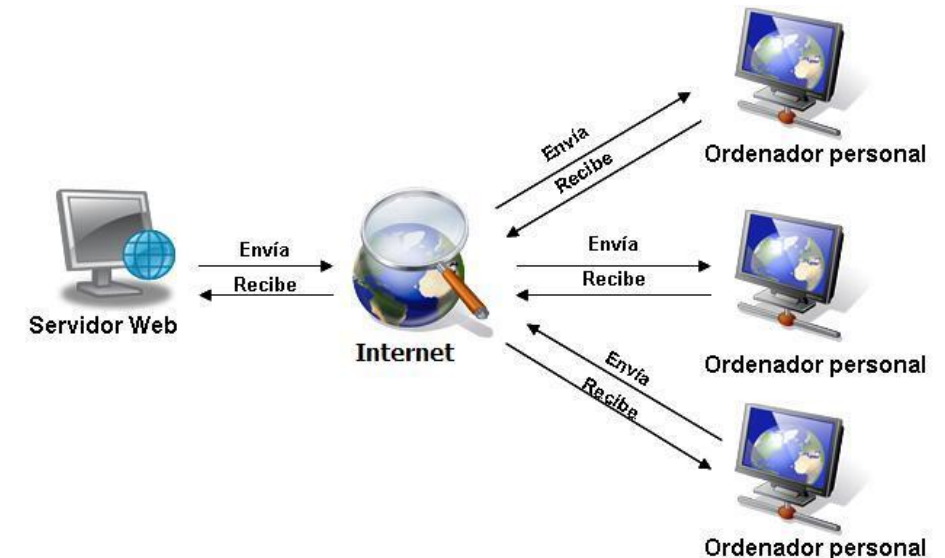
- **Servicio de Internet**
  - Los servicios de Internet permiten a los usuarios acceder a la información de Internet.
    - Otros servicios de Internet: FTP, correo
- Desarrollado en 1989 por Tim Berners-Lee
- Este servicio se basa en 3 pilares:





### LA WEB

- Con un **navegador** web, un usuario visualiza **sitios web** que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y **navega** a través de ellas usando **hiperenlaces**.
  - El **sitio web** se compone de **páginas web** y está alojado en un **servidor web**.
- Las páginas web tienen formato **HTML** y **CSS**, tienen enlaces a otras páginas y a contenido multimedia (imágenes, vídeos, audio)
- El **protocolo** con el que se comunican el cliente y el servidor es Hiper Text Transfer Protocol (**HTTP**)
- Las páginas se **identifican** con un nombre único llamado coloquialmente dirección web (**URL**)



El proceso de comunicación según HTTP





### LA WEB

**URL: protocolo://maquina:puerto/camino/fichero ?**

**http://java.org:4040/ejemplo/inicio.html**

#### ➤ **Protocolo**

- Generalmente es **http** o **https** (HyperText Transport Protocol Secure)
- Pueden emplearse otros protocolos como ftp (File Transfer Protocol)

#### ➤ **Máquina**

- Nombre o la IP del servidor al cual nos queremos conectar.
- Se suele emplear un nombre (Google.com) que es traducido a una IP por el servicio de DNS.

➤ **Puerto (opcional).** Por defecto, cada protocolo tiene un puerto asignado, http tiene asignado el puerto 80 y https el puerto 443.

➤ **Camino:** ruta en el sistema de archivos de la máquina remota donde se encuentra el recurso al cual queremos acceder. Dicha ruta es relativa al directorio raíz de la web.

➤ **Fichero** es el recurso concreto al que queremos acceder dentro de un directorio de la máquina.





### PÁGINA WEB

- Una página web es un **documento** accesible desde cualquier **navegador** con acceso a internet, y que puede incluir **audio, vídeo, texto** y sus diferentes combinaciones.
- Generalmente es un fichero con extensión **.html**.
- Puede contener información de estilos (colores, tipos de letra, distribución de elementos,...) en el formato **CSS**.







### SITIO WEB

- Un sitio web es una **colección** de **páginas web** relacionadas entre sí
- El conjunto de páginas web que forman un sitio web suelen compartir la primera parte de la dirección web (el dominio)
  - <https://educa.aragon.es>
  - <http://es.wikipedia.org>
- Tipos de sitios web
  - Web corporativas de empresas
  - Blogs, redes sociales
  - Páginas personales
  - Buscadores
  - Sitios de noticias
  - Enciclopedias



# APLICACIONES WEB

## UD1.- Introducción a la web

### APLICACIÓN WEB

En los comienzos...

- Todos los sitios web eran conjuntos de páginas web en forma de **ficheros HTML**
- Los sitios web eran como libros pero con navegación mediante **enlaces**
- A estas páginas se las denominaba páginas **web estáticas**

Actualidad → **Aplicaciones web dinámicas**

- Además de los ficheros HTML las aplicaciones contienen **programas** que se ejecutaban cada vez que un usuario solicita una página
  - Se utilizan lenguajes de script como PHP





### APLICACIÓN WEB

## Aplicación Web

- Aquella **aplicación** que los usuarios pueden utilizar accediendo a un **servidor web** a través de **Internet** mediante un **navegador**
- Es una **aplicación software** cuya interfaz de usuario se usa desde un navegador web
- ¿ Por qué son tan populares?
  - El navegador web como cliente ligero
  - Independencia del sistema operativo
  - Facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios
- Generalmente nos referimos a ellas como **Web**
- La mayor parte de las páginas web actuales son dinámicas



### APLICACIÓN WEB

## Diferencia entre las Apps y las aplicaciones webs

- Apps
  - Tienen que ser instaladas en un dispositivo y dependen de una store.
  - Ocupan espacio en el almacenamiento
  - Proporcionan una mayor facilidad de acceso
- Una aplicación web no requiere instalación previa





### 2. EVOLUCIÓN HISTÓRICA DE LA WEB

➤ **La Web** fue desarrollada en 1989 por Tim Berners-Lee y se publico en 1992.

- **Web:** sistema que funciona a través de internet, por el cual se pueden transmitir diversos tipos de datos a través del Protocolo de Transferencia de Hipertextos o HTTP.

➤ **Web 1.0 (1993-2003)**

- Término utilizado para referirnos a la primera etapa de la Web
- Sitio web **estáticos** simples → puramente informativos
- El nacimiento de google, las wikis, y los blogs nos lleva a la Web 2.0



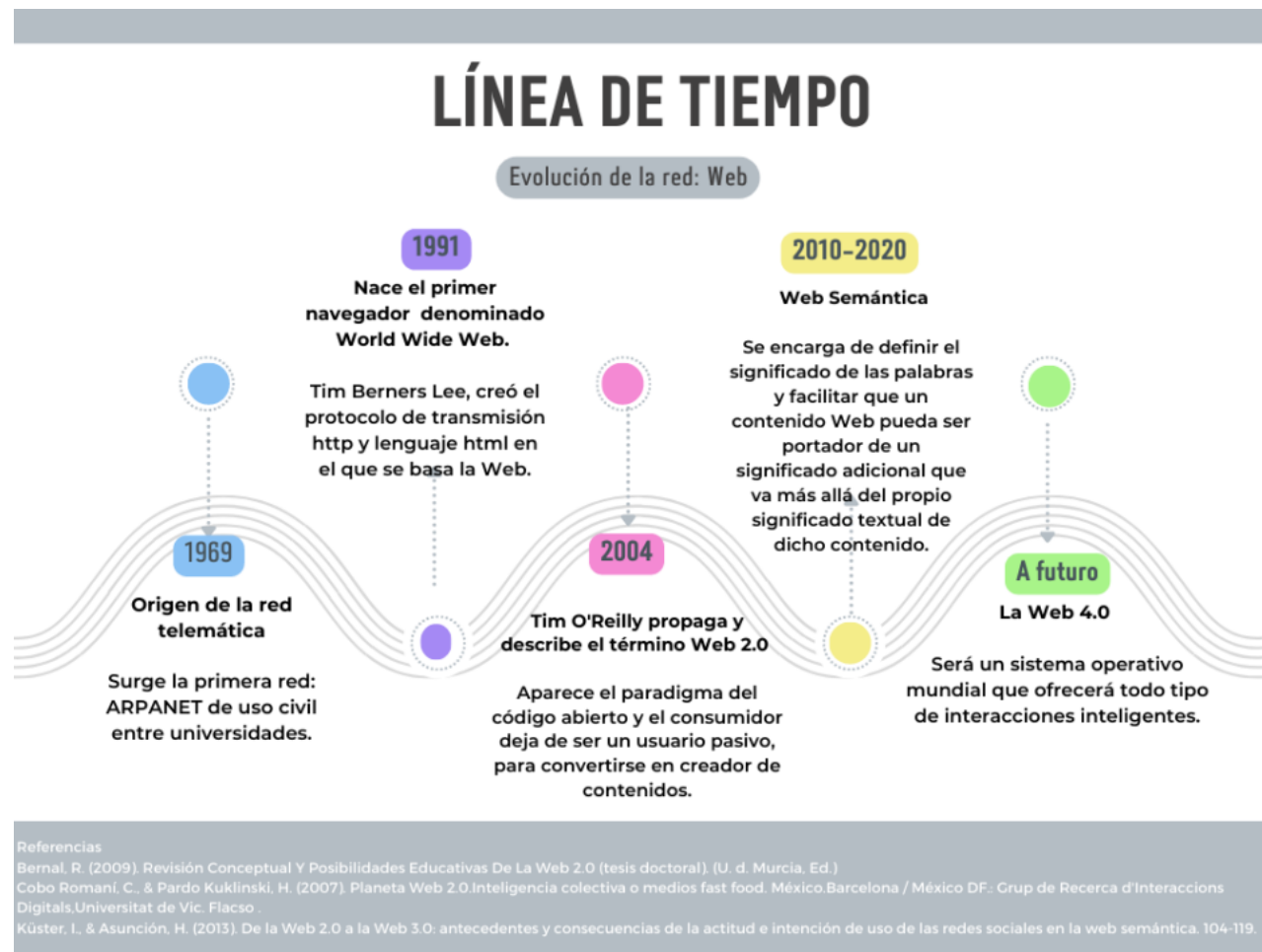
➤ **La Web 2.0 o Web social**

- Término acuñado por Dale Dougherty (O'Reilly) en 2004
- Nueva generación de sitios web que permiten a las personas colaborar y compartir información en línea de formas que antes no eran posibles
- itios web **dinámicos**, en los que al usuario se le permita interactuar, generar contenido, o formar parte de comunidades virtuales.





### 2. EVOLUCIÓN HISTÓRICA DE LA WEB





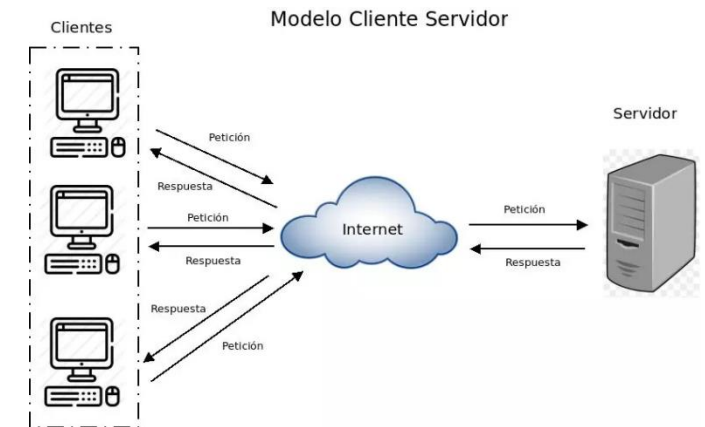
### 3. NAVEGADORES Y SERVIDORES WEB

#### ➤ La web sigue una arquitectura **cliente-servidor**

- El **navegador** actúa como **cliente**
- El **servidor web** actúa como **servidor**
- La comunicación se establece usando el protocolo **http**

#### ➤ El usuario escribe una dirección web (**URL**) en el navegador

- La dirección contiene el nombre del servidor web y el nombre del recurso que solicita
- El navegador hace una **petición** al servidor y solicita el recurso
- El navegador **descarga** el recurso y lo visualiza o lo descarga
- Si el **recurso** es un página HTML, además de visualizar su contenido, descarga los recursos adicionales ( imágenes, estilos, ...) y los **visualiza**







### SERVIDOR WEB

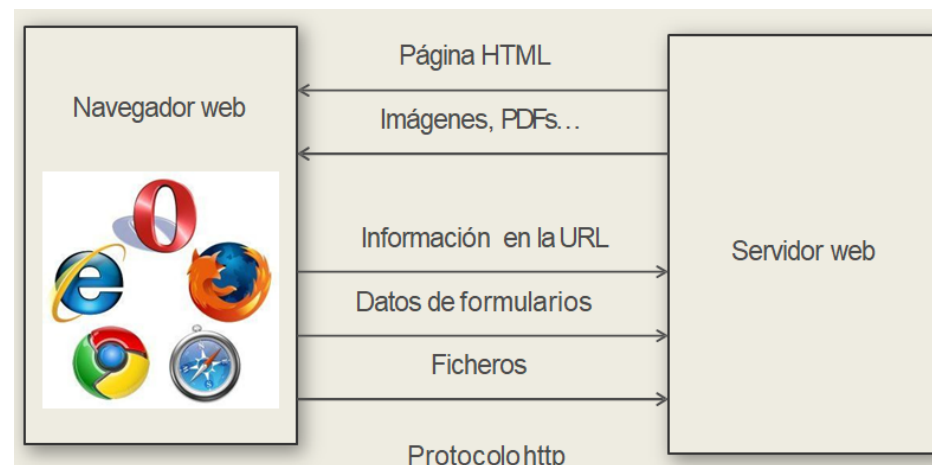
#### Compuesto por:

##### ➤ Software

- Alberga sitios web estáticos y los sirve con el protocolo http a los navegadores web
- Ejecuta aplicaciones web que el usuario utiliza desde un navegador

##### ➤ Hardware

- Servidor físico en el que se ejecuta el software del servidor web
- Dispone de otros servicios adicionales como un sistema gestor de bases de datos







### SERVIDOR WEB

- Un **servidor web** básico sirve por http los ficheros que están en el **disco duro**
- Puede servir **cualquier tipo de fichero**, aunque lo habitual son los ficheros que un **navegador** reconoce (html, jpg, png, pdf...)
- Cuando recibe una petición, devuelve el **fichero** del disco duro que se ajuste a la ruta indicada en la URLs
- La mayoría de los **servidores web** permiten que en cada petición se ejecute un **programa** que genera **dinámicamente** el recurso que se envía al usuario (**server-side scripting**)
- A este contenido generado “al vuelo” se le denomina **contenido dinámico**, en contraposición al contenido estático obtenido del disco duro
- Es bastante habitual que el contenido dinámico se genere con la información de una **base de datos**
- Los servidores web también pueden **procesar información** que les llega del mismo
  - Datos contenidos en formularios web ...
  - Esta funcionalidad permite el desarrollo de **aplicaciones web** completas

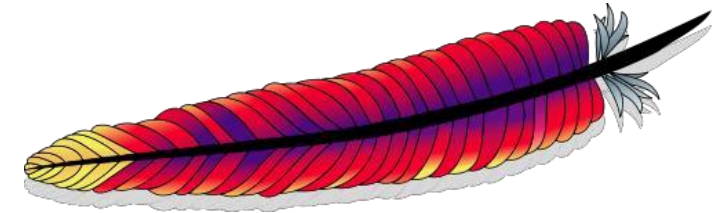


### SERVIDOR WEB

## Servidores web más utilizados

### ➤ Apache

- Permite ejecutar contenido dinámico
- Es el servidor más popular
- Es multiplataforma
- <http://httpd.apache.org>



### ➤ NginX

- Servidor multiplataforma y licencia libre (tipo BSD)
- Permite ejecutar contenido dinámico
- Es muy rápido y se usa como balanceador de carga
- <http://nginx.net/>

# NGINX



### SERVIDOR WEB

#### Web Servers

##### Most popular web servers

© W3Techs.com	usage	change since 1 August 2018
1. Apache	45.6%	-0.3%
2. Nginx	39.6%	+0.6%
3. Microsoft-IIS	9.4%	-0.1%
4. LiteSpeed	3.4%	+0.1%
5. Google Servers	0.9%	-0.1%

percentages of sites

##### Fastest growing web servers since 1 August 2018

© W3Techs.com	sites
1. Nginx	1,058
2. LiteSpeed	68
3. Apache Traffic Server	60

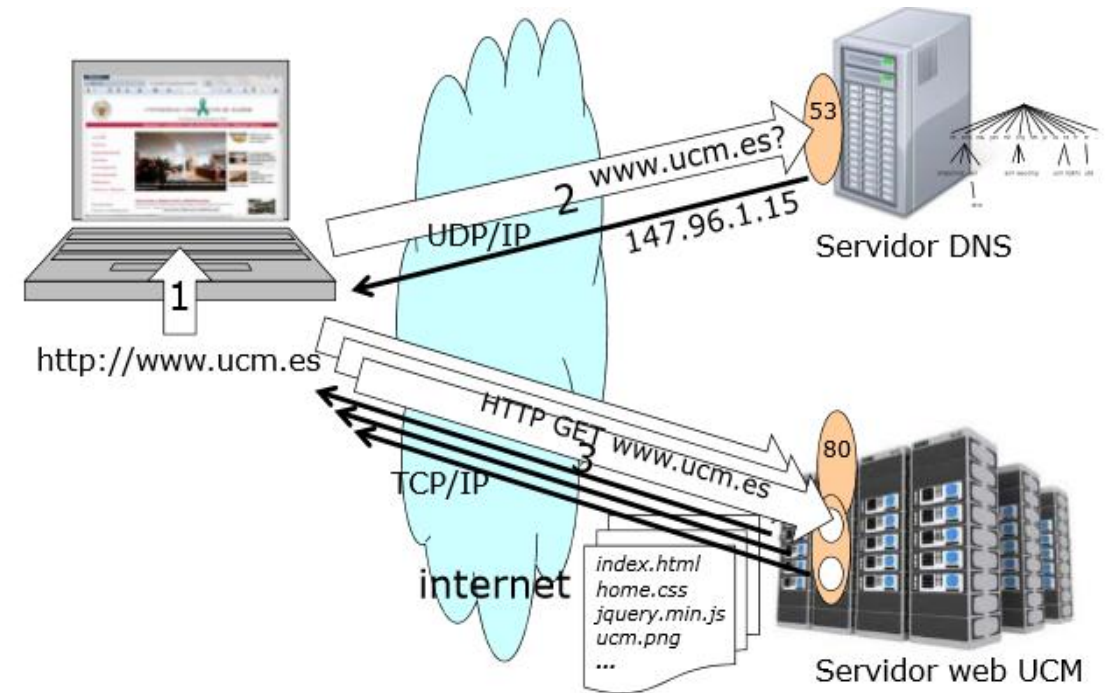
daily number of additional sites  
in the top 10 million



### PROTOCOLO HTTP

#### Protocolo HypertextTransfer Protocol estándar (W3C y IETF)

- El puerto por defecto es el **80**
- Es un protocolo cliente servidor
  - Los navegadores (clientes) hacen peticiones a los servidores web
- La petición solicita un recurso, identificado por la **URL** (Uniform Resource Locator)
- Es un protocolo **sin estado**
  - No guarda ninguna información sobre conexiones anteriores





### PROTOCOLO HTTP

#### Ciclo de vida de una petición http

- En un navegador se introduce la dirección web

<http://www.host.com/index.html>

- El navegador abrirá una conexión al puerto 80 del servidor (cuya IP obtiene por DNS)

[www.host.com](http://www.host.com)

- Enviará la petición con formato

```
GET /index.html HTTP/1.1 Host:
www.example.com User-Agent: tipo
navegador [Línea en blanco]
```

- El servidor devolverá una respuesta con el formato

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html>
<body>
<h1>Página principal</h1>
(Contenido)...
```



### PROTOCOLO HTTP

#### Ciclo de vida de una petición http

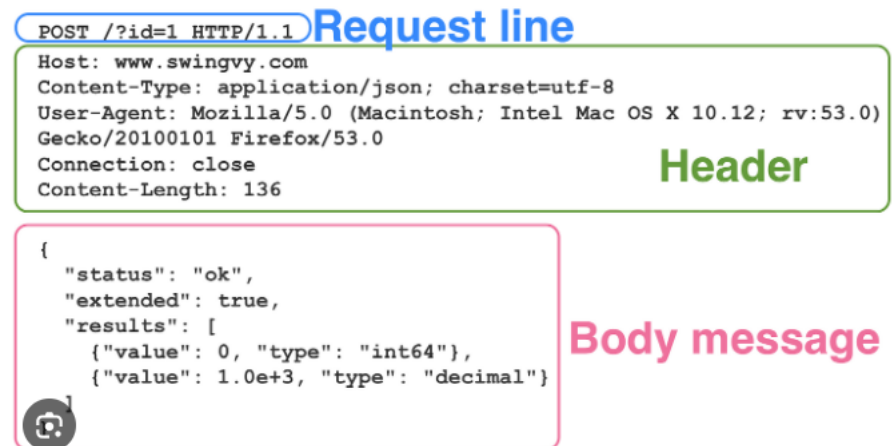
- El navegador analiza la **respuesta** del servidor.
- Si es correcta (**200 OK**) se lee el contenido
- Se determina el tipo de contenido por el “**Content-Type**”. En este caso, “text/html”
- Se lee el contenido y se **muestra** en el navegador
- Si el contenido referencia a más contenido (imágenes, reglas de estilo, etc...) se vuelve a hacer una petición http por cada uno de ellos



## PROTOCOLO HTTP

### Mensaje HTTP request

- La primera parte de la petición es el método que se emplea para realizar la petición (GET o POST)
  - GET. Devuelve el recurso identificado en la URL pedida.
  - POST. Indica al servidor que se prepare para recibir información del cliente. Suele usarse para enviar información desde formularios.





### PROTOCOLO HTTP

**GET** /en/html/dummy?name=MyName&married=not+single&male=yes HTTP/1.1

Host: www.explainth.at

User-Agent: Mozilla/5.0 (Windows;en-GB; rv:1.8.0.11) Gecko/20070312

Firefox/1.5.0.11

Accept: text/xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5

Accept-Language: en-gb,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7

Keep-Alive: 300

Connection: keep-alive

**POST** /en/html/dummy HTTP/1.1

Host: www.explainth.at

User-Agent: Mozilla/5.0 (Windows;en-GB; rv:1.8.0.11) Gecko/20070312

Firefox/1.5.0.11

Accept: text/xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5

Accept-Language: en-gb,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://www.explainth.at/en/misc/httpreq.shtml

Content-Type: application/x-www-form-urlencoded

Content-Length: 39

**name=MyName&married=not+single&male=yes**





## PROTOCOLO HTTP

### Mensaje HTTP response

- Una respuesta del servidor en el protocolo http sigue la siguiente estructura:

Versión-http SP código-estado SP frase-explicación CRLF

(nombre-cabecera: valor-cabecera ("," valor-cabecera)\* CRLF)\*

Cuerpo del mensaje

- El **código de estado** es un código que indica si la petición ha tenido éxito o habido algún error con ella. La frase de explicación suele proporcionar alguna explicación del error.
- Los códigos de estado del protocolo http son números de tres dígitos que forman parte de las respuestas http. Estos códigos explican qué ha sucedido al intentar llevar a cabo una petición.



### PROTOCOLO HTTP

*HTTP/1.1 200 OK*

*Content-Type: text/xml; charset=utf-8*

*Content-Length: length*

*<html>*

*<head> <title> Título de nuestra primera página </title> </head>*

*<body>*

*¡Hola mundo!*

*</body>*

*</html>*



### PROTOCOLO HTTP

#### **Códigos 1xx : Mensajes**

- o 100-111 Conexión rechazada

#### **Códigos 2xx: Operación realizada con éxito**

- o 200 OK
- o 201-203 Información no oficial
- o 204 Sin Contenido
- o 205 Contenido para recargar
- o 206 Contenido parcial

#### **Códigos 3xx: Redirección**

- o 301 Mudado permanentemente
- o 302 Encontrado
- o 303 Vea otros
- o 304 No modificado
- o 305 Utilice un proxy
- o 307 Redirección temporal

#### **Códigos 4xx: Error por parte del cliente**

- o 400 Solicitud incorrecta
- o 402 Pago requerido
- o 403 Prohibido
- o 404 No encontrado
- o 409 Conflicto
- o 410 Ya no disponible
- o 412 Falló precondition

#### **Códigos 5xx: Error del servidor**

- o 500 Error interno
- o 501 No implementado
- o 502 Pasarela incorrecta
- o 503 Servicio no disponible
- o 504 Tiempo de espera de la pasarela agotado
- o 505 Versión de HTTP no soportada