UNIVERSIDAD DE GRANADA

Programación Web

DECSAI

TEMA 1. INTRODUCCIÓN

Curso 2018-2019

Contenido

- Internet y la web
- Modelo cliente-servidor
- Desarrollo para Web
- Arquitecturas Web

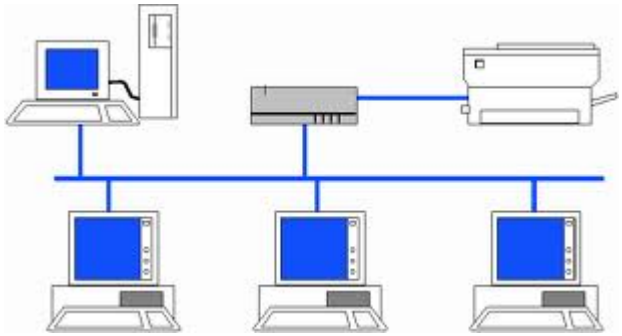
INTERNET Y LA WEB

Ordenador

- Máquina electrónica para procesar datos.
- Capacidades básicas:
  - Cálculo
  - Almacenamiento
  - Comunicación

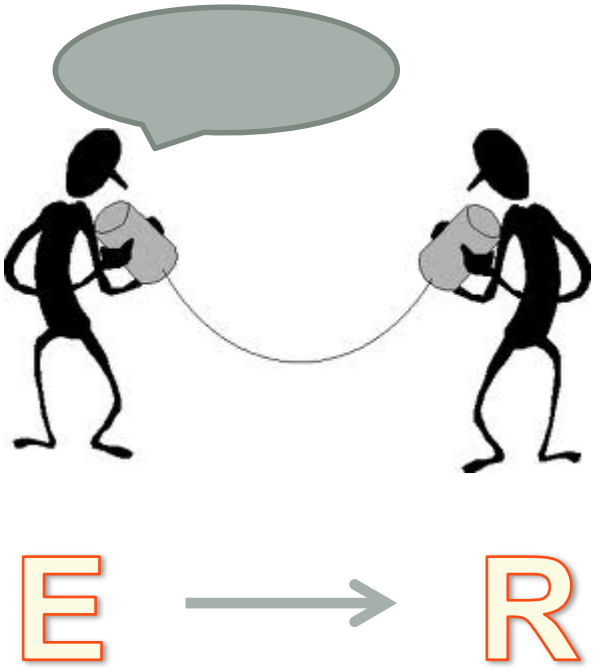
Red de ordenadores

- Conjunto de equipos informáticos interconectados mediante dispositivos físicos para el transporte de datos con el objeto de compartir información, recursos y ofrecer servicios.

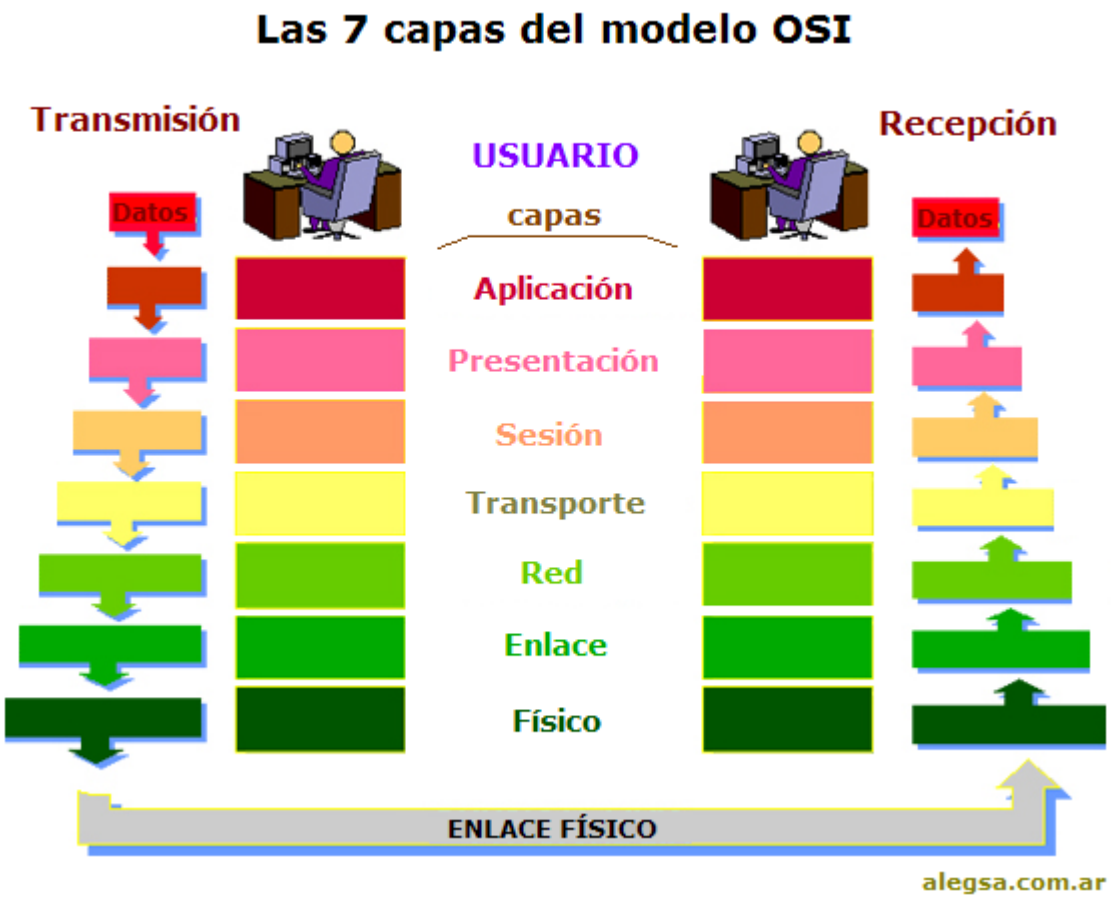


Red de ordenadores: comunicación

- Comunicación:
  - Emisor
  - Receptor
  - Medio
  - mensaje



Modelo OSI

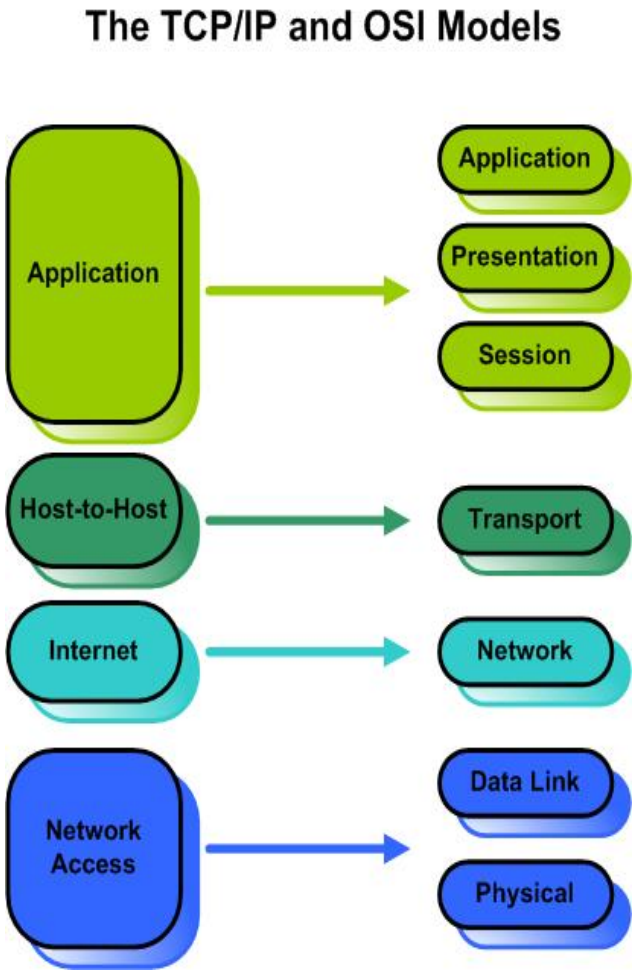

Las 7 capas del modelo OSI
Transmisión
Recepción
USUARIO
capas
Datos
Datos
Aplicación
Presentación
Sesión
Transporte
Red
Enlace
Físico
ENLACE FÍSICO
alegsa.com.ar

Protocolo de comunicaciones

- Conjunto de reglas normalizadas para la representación, señalización, autenticación y detección de errores necesario para enviar información a través de un canal de comunicación.
- **Protocolo** (informático): protocolo para la comunicación entre ordenadores

Protocolos TCP/IP

- Familia de varios protocolos donde destacan TCP e IP.
- TCP: Transmission Control Protocol,
- IP: Internet Protocol



Otros protocolos TCP/IP

- HTTP: HyperText Transfer Protocol
- FTP: File Transfer Protocol
- SMTP: Simple Mail Transfer Protocol
- IMAP: Internet Message Access Protocol
- SSH: Secure Shell, protocolo

Internet

- Conjunto descentralizado de redes de comunicación interconectadas que usan los protocolos de la familia TCP/IP
- Es previo a la web
- Hay muchos servicios definidos sobre Internet
- No está controlada por ningún país

World Wide Web

- También llamada Web.
- Sistema de distribución de información basada en hipertexto o hipermedios enlazados y accesibles a través de Internet.
- Creada por Tim Berners-Lee en 1989

Un poco de historia

- Idea subyacente: V. Bush en los 40
- Primer sistema de hipertexto: T. Nelson en los 50
- 1980: Tim Berners-Lee propone ENQUIRE
- 1989: Propuesta redactada
- 1991: el proyecto se publica en newsgroups, con ayuda de R. Cailliau
- 1993: el CERN declara gratuita la web
- 1993: aparición del navegador con GUI: Mosaic

Términos

- Página web
- Lenguajes de marcado
- Servidor web
- Navegador
- (hiper)Enlace
- URL
- Directorio
- Buscador

Estándares web

- URI (Uniform Resource Identifier): sistema universal para hacer referencia a recursos en la web.
- HTTP (HyperText Transfer Protocol): protocolo de comunicación entre navegador y servidor
- HTML (HyperText Markup Language): Lenguaje de marcado de hipertexto, usado para definir la estructura y contenido de los documentos
- XML (eXtensible Markup Language): usado para describir la estructura de documentos

URI

- Cadena de caracteres que identifica unívocamente un recurso.
- Tipos:
  - URL: Localizador uniforme de recurso: Indican la ubicación exacta del recurso (Dirección)
  - URN: Universal Resource Name. No indican dónde se encuentra el recurso

Partes de un URI

- http://sci2s.ugr.es/dicits/index.php?p=software#desc
- Esquema: http
- Autoridad: sci2s.ugr.es
- Ruta: dicits/index.php
- Consulta: p=software
- Fragmento: #desc

HTML

- Lenguaje de marcado escrito en forma de etiquetas, flanquedas por ángulos: <>
- Describe estructura
- Puede definir aspectos de apariencia
- Puede incluir comportamiento dinámico, a través de *scripts*
- Puede incluir contenidos de tipo MIME
- Versión actual: HTML5

XML

- Derivado del SGML, permite definir la gramática de lenguajes para estructurar grandes documentos.
- Se puede utilizar para intercambio de datos en Internet, entre bases de datos, editores de texto, hojas de cálculo, … (P. ej. Open Office XML)
- Tecnología sencilla complementada con otras

WWW Consortium (W3C)

- Comunidad internacional donde las organizaciones miembro, la plantilla y el público en general trabajan para el desarrollo de protocolos para la web.
- *Misión*: liderar el WWW para alcanzar todo su potencial mediante el desarrollo de protocolos y guías que garanticen el crecimiento de la web.
- *Principios*: estándares abiertos, web para todos, web para todo; web de datos, servicios, confianza

www.w3.org

MODELO CLIENTE-SERVIDOR

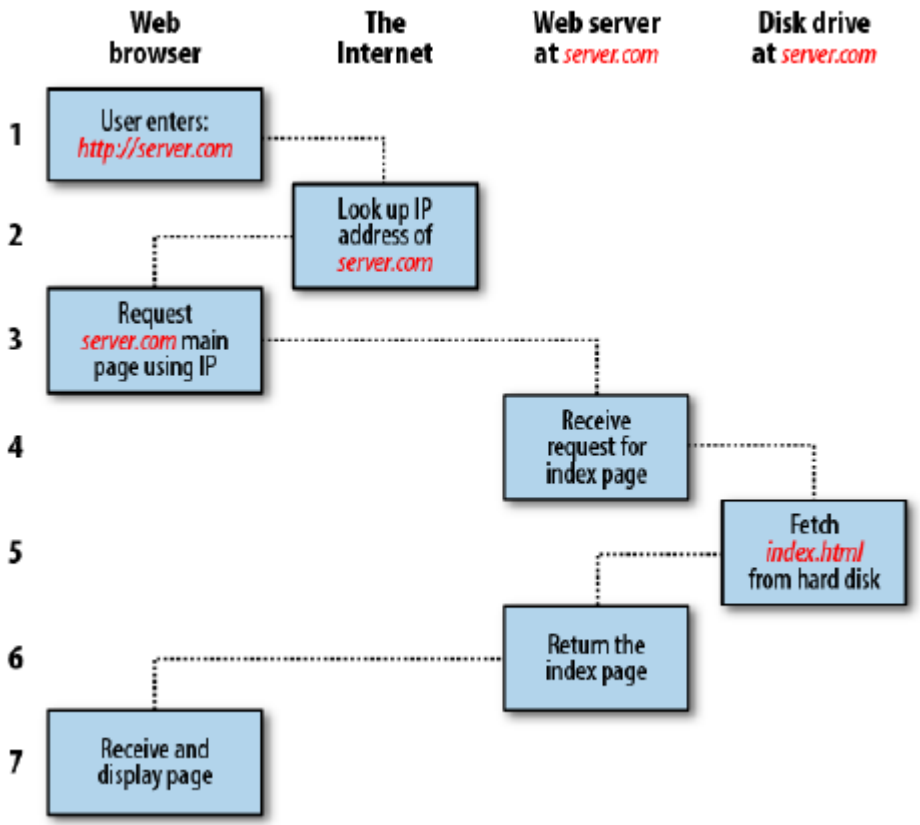
Arquitectura cliente-servidor (C-S)

- Modelo de aplicación distribuida donde las tareas se reparten entre los proveedores de recursos o servicios (servidores) y los solicitantes (clientes)
- Separación conceptual de tipo lógico, que facilita el diseño y la implementación
- Alternativa a:
  - Arquitectura monolítica
  - Redes entre pares

Arquitectura C-S en la web

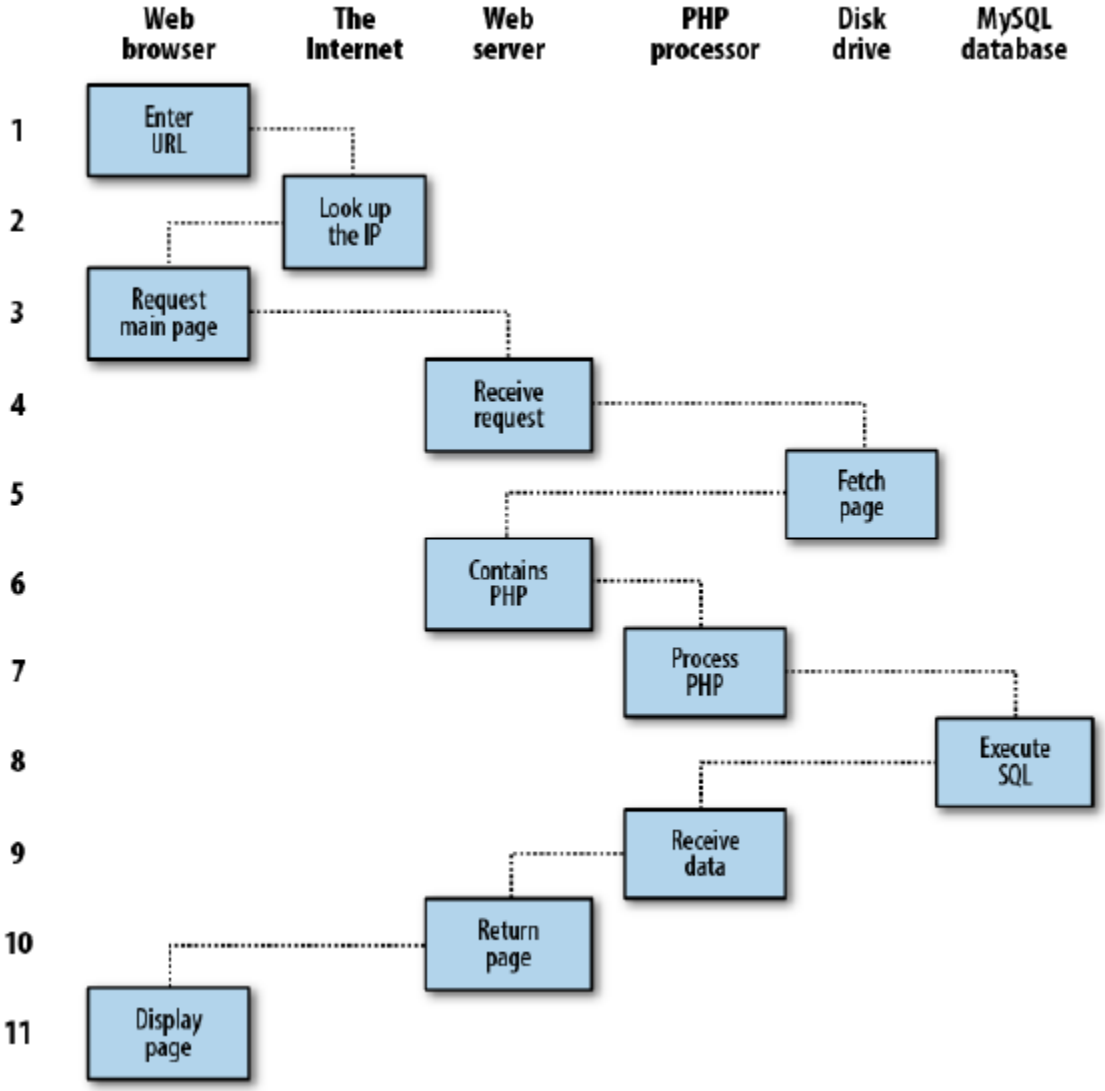
- *Servidor* Web: presta servicios tras peticiones realizadas mediante http
- *Cliente*: navegador (u otro programa) que envía peticiones al servidor web; interpreta el contenido recibido

Proceso de servicio de página (estático)



What happens when you type an URL in the browser

Servicio de contenido dinámico



Servidores web

- Puede hacer referencia al hardware, software o ambos.
- Sistema que proporciona contenido en la web respondiendo al protocolo HTTP.
- Dispensar páginas web y contenido adicional: imágenes, hojas de estilo, scripts.

Estadísticas de uso (software)

Web Servers

Most popular web servers

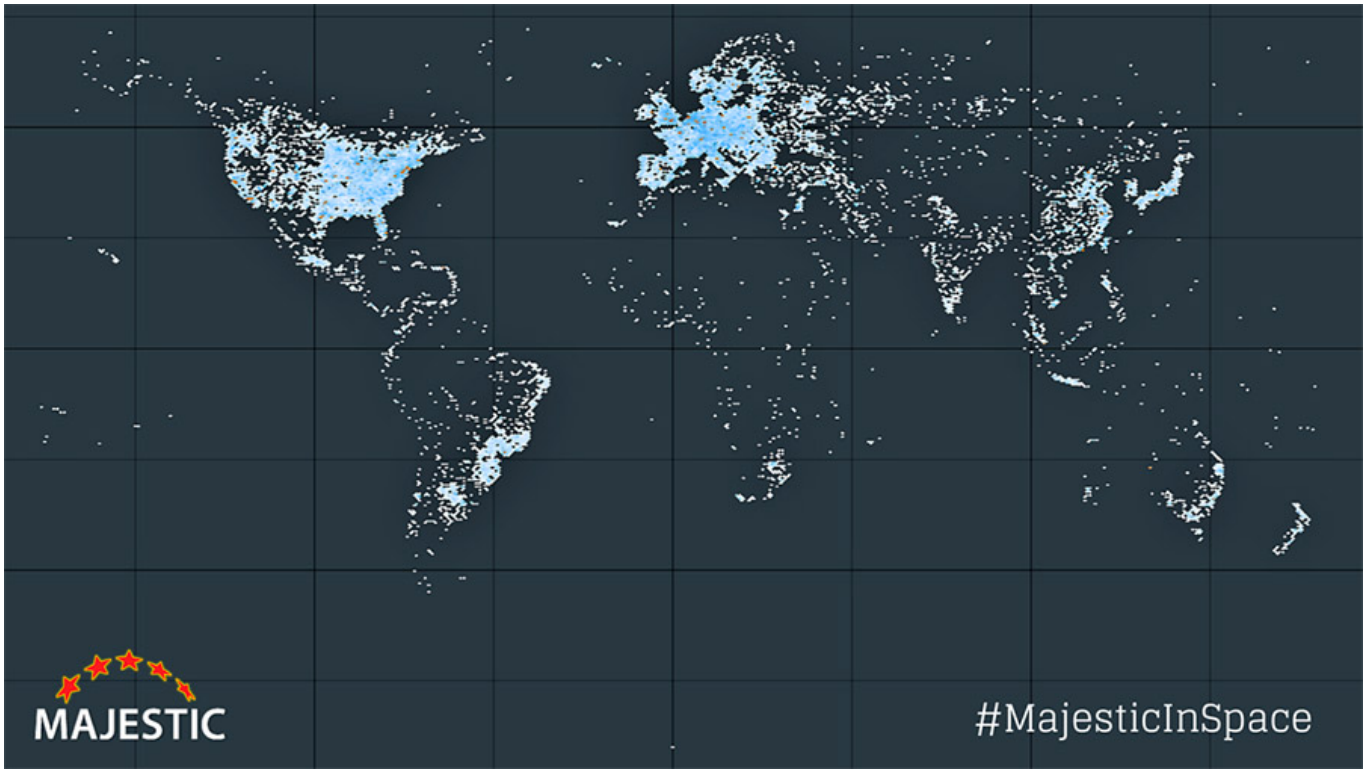
© W3Techs.com	usage	change since 1 January 2019
1. Apache	44.2%	-0.4%
2. Nginx	41.1%	+0.4%
3. Microsoft-IIS	8.8%	-0.2%
4. LiteSpeed	3.9%	+0.2%
5. Google Servers	0.9%	

percentages of sites

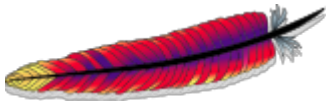
w3techs.com

Servidores web sobre mapamundi

MicroSiervos: servidores web sobre el globo terráqueo



Apache



- Servidor web de código abierto multiplataforma, altamente configurable, extensible y modular.
- Apareció en 1995, como *parches* sobre el NCSA httpd.
- Desde abril de 1996 es el servidor más usado de internet. Alcanzó el 70% de cuota de uso.
- Su desarrollo derivó en la Apache Software Foundation

Apache en Linux

- Conseguir un servidor físico/virtual con Linux (p.ej. CentOS, ubuntu)
- Instalar apache
- Configurar apache
- Activar servidor
- Incluir contenido

Usar contenedores (docker)
o máquinas virtuales

Otros servidores web

- Nginx
- Cherokee
- Tornado
- Lighttp
- IIS
- Listado de wikipedia

¡Recomendación!:
Instalar distintos servidores web y
ensayar configuraciones distintas

Navegador web

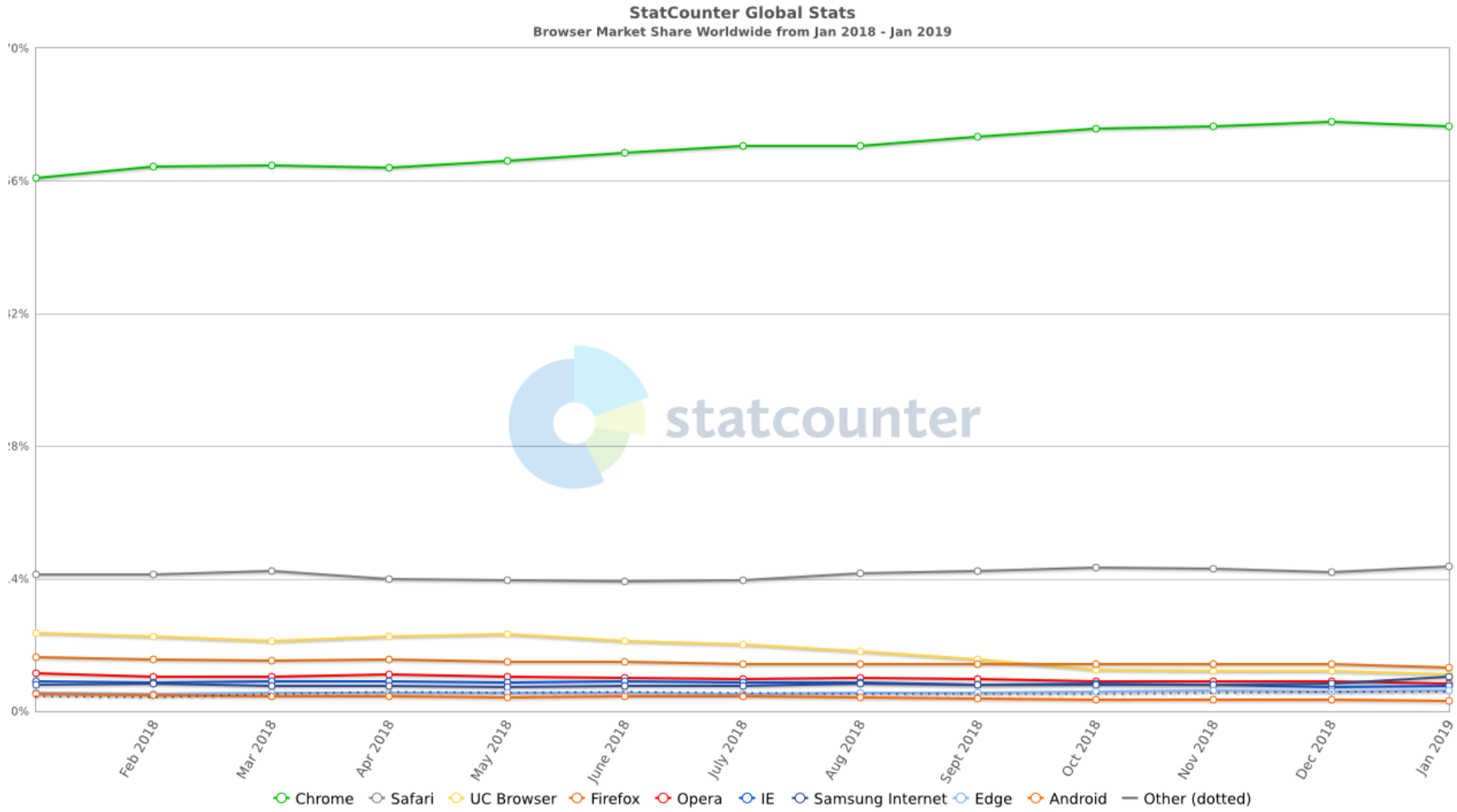
- (Browser): aplicación para obtener y mostrar información de la web
- Debe visualizar los documentos obtenidos, interpretando su estructura y mostrando los recursos multimedia incrustados
- Debe facilitar la navegación accediendo a hiperenlaces incluidos en los documentos

Inside look at modern web browser

Navegadores web

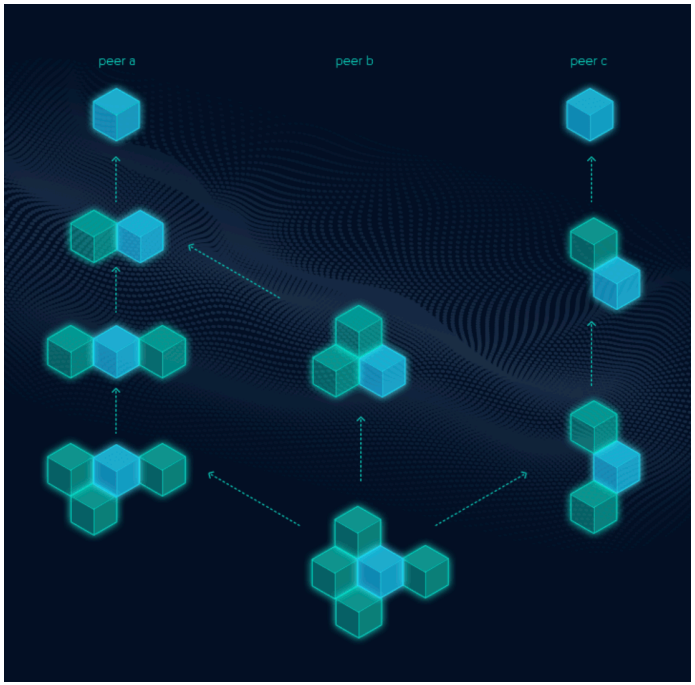
- Google Chrome/Chromium
- Firefox/Mozilla
- Internet Explorer/Edge
- Opera
- Safari
- Lynx
- Konqueror
- Epiphany
- *TorBrowser*
- …

Estadísticas de uso



Web descentralizada: PeerPad.net

- Editor colaborativo en tiempo real para desarrollar una web descentralizada (sobre IPFS).



Interplanetary File System:
A peer-to-peer hypermedia protocol to make teh web faster, safer and more open

DESARROLLO WEB

Desarrollo para web

- Término extenso que engloba las tecnologías y procesos de creación de páginas web (o *web sites*) y aplicaciones web
- El ámbito incluye desde simples páginas estáticas hasta aplicaciones de comercio electrónico, aplicaciones ofimáticas, aplicaciones de gestión y contabilidad o redes sociales
- Desarrollo de aplicaciones móviles o para Cloud Computing

Aspectos

- Diseño: estructura, contenido informativo, aspecto y funcionalidad
- Desarrollo en el ámbito del servidor
- Desarrollo en el ámbito del cliente
- Entornos de trabajo para desarrollo
- Bases de datos
- Interfaz de usuario

Diseño de sitios web

- Cuidar la interacción con el usuario
  - Colores
  - Fondos
  - Fuentes
  - Procesos de búsqueda
  - Distribución del contenido en cada página
- Consideraciones sobre accesibilidad
- Compatibilidad con navegadores

Tecnologías en el ámbito del servidor

- PHP
- ASP
- CGI (perl, python, C, …)
- Java
- .NET
- Python
- Ruby
- Node.js
- Scala
- Extensiones de lenguajes: R, clojure, …

Tecnologías en el ámbito del cliente

- JavaScript
- AJAX
- Flash
- jQuery
- HTML5
- CSS3
- webassembly
- …

Marcos de trabajo para desarrollo

- Zend (PHP)
- Django (Python)
- Ruby on Rails (Ruby)
- Websphere
- …

Sistemas Gestores de Bases de Datos

- MySQL
- PostgreSQL
- DB2
- Oracle
- SQLite
- Redis
- MongoDB
- CouchDB
- Cassandra, …

- Conexión estandarizada:
  - ODBC
  - JDBC

PROTOCOLO HTTP

Protocolo HTTP

- Protocolo de comunicación (a nivel de aplicación) basado en TCP/IP para el acceso a contenidos (ficheros, datos, recursos multimedia, …) a través de la Web
- Es un protocolo sin conexiones
- Depende del formato del contenido. Catálogo definido por la especificación MIME
- Protocolo sin estado. No se mantiene información entre distintas solicitudes
- En 2015 se estableció el estándar HTTP/2

Estructura del mensaje

- Los **mensajes** se representan como texto.
- Las peticiones y respuestas intercambiadas en HTTP tienen la siguiente estructura:
  - Línea inicial finalizada con CRLF
  - Varias líneas de cabecera finalizadas por CRLF
  - Línea en blanco, CRLF
  - Cuerpo del mensaje

Línea inicial (para petición)

- Tiene tres partes, separadas por espacios
  - Nombre de método HTTP que se invoca
  - Trayectoria local del recurso solicitado
  - Versión de HTTP que se usa
- Ejemplo:

```
GET /path/to/file/index.html HTTP/1.0
```

Línea inicial (para respuesta)

- Tiene tres partes, separadas por espacios:
  - Versión de HTTP
  - Código de estado respuesta
  - Una frase en inglés descriptiva del código de respuesta
- Ejemplo:
  - `HTTP/1.0 200 OK`
  - `HTTP/1.0 404 Not Found`

Líneas de cabecera

- Dan información sobre la solicitud o respuesta, o sobre el objeto enviado en el mensaje del mensaje
- Header-Name: <valor> CRLF
- No sensible a mayúsculas
- Si hay espacios al principio de la línea, es continuación de la línea previa
- Ejemplo:
  - `User-agent: Mozilla/3.0Gold`
  - `Last-Modified: Fri, 31 dec 2012 23:10:00`

El cuerpo del mensaje

- Ubicación con la respuesta a la solicitud
- A veces hay líneas en la cabecera que describen el contenido del cuerpo:
  - `Content-Type`: indica el tipo MIME, text/html o image/gif
  - `Content-Length`: número de bytes en el cuerpo

Métodos de HTTP

- GET: recuperar información identificada por el URI
  - Modificador para solicitar sólo si ha habido actualización: encabezado If-Modified-Since
  - Se usa para enviar formularios
- HEAD: Igual que GET, pero sólo pide las líneas de cabecera, no el contenido
- POST: Enviar datos al servidor, para su procesamiento.
  - Mensaje con cuerpo
  - El URI especifica un programa o módulo de procesamiento
  - La respuesta suele ser calculada, no contenido estático

Ejemplo de POST

- POST /camino/al/programa HTTP/1.0
- FROM: usuario@ugr.es
- User-Agent: Firefox/1.0
- Content-Type: application/x-form-urlencoded
- Content-Length: 32

- Home=Libro&favorito+reciente=espasa

Cabeceras más habituales

- Allow
- Authorization
- Content-Encoding
- Content-Length
- Content-Type
- Date
- Expires
- From
- If-Modified-Since
- Location
- Pragma
- Referer
- Server
- User-Agent
- WWW-Authenticate

Códigos más habituales

- 100 Continue
- 200 Ok
- 201 Created
- 202 Accepted
- 204 No Content
- 300 Multiple choices
- 301 Moved permanently
- 302 Found
- 303 See Other
- 305 Use Proxy
- 400 Bad Request
- 401 Unathorized
- 403 Forbidden
- 404 Not Found
- 405 Method not Allowed
- 408 Request Timeout
- 500 Internal Server Error
- 503 Service Unavailable

Ejemplo: http://www.host.com/camino/fichero.html

```
GET /camino/fichero.html HTTP/1.0
From: usuario@ugr.es
User-Agent: Firefox/15.0
<línea en blanco>
```

Respuesta del servidor

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 2011 22:00:00 GMT
Content-Type: text/html
Content-Length: 2479
<línea en blanco>
<html>
<body>
<hl>Feliz Año Nuevo</hl>
…
</body>
</html>
```

Probando un servidor desde línea de órdenes

```
telnet www.servidor.com 80

GET /camino/fichero.html HTTP/1.0
<cabeceras>
<línea en blanco>
```

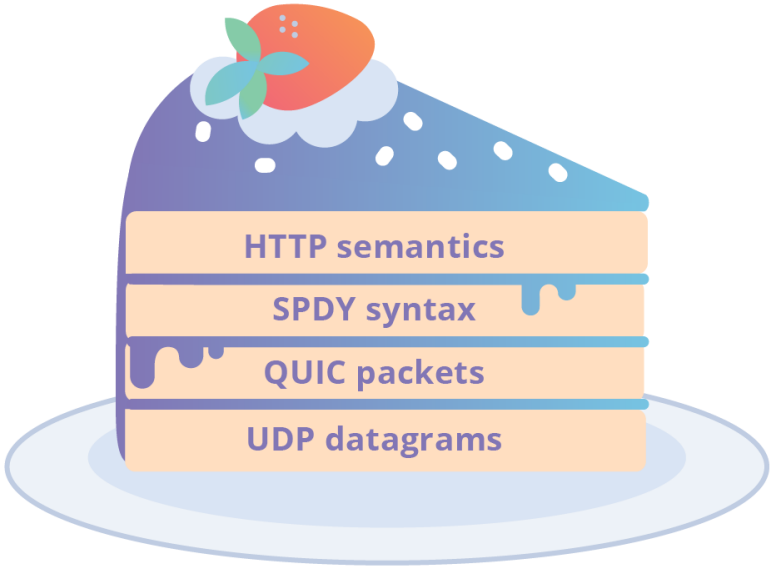
HTTP/2

- Revisión del protocolo HTTP a partir de la propuesta SPDY. (2015).
- Objetivos:
  - Crear mecanismo de negociación
  - Mantener compatibilidad con HTTP1.1
  - **Reducir la latencia** en la descarga de páginas:
    - Una **única conexión** para todos los elementos de una página web.
    - Compresión de datos en las cabeceras
    - Protocolo binario
    - Incluir **Server Push** (enviar más datos antes de que el cliente los pida)
    - “Pipelining” de peticiones
    - **Multiplexación** de peticiones: enviar y recibir varios mensajes en paralelo

HTTP/3

- HTTP over QUIC
- Mejorando la velocidad para una web más segura
- Aplicación de HTTP para conectarse con la capa de transporte QUIC

HTTP3: From root to tip



ARQUITECTURAS WEB

Diseño de arquitectura web

- Disciplina cuyo objeto es la planificación y diseño de sitios y aplicaciones web. Incluye los aspectos técnicos, estéticos y funcionales. En particular, de organización y gestión de contenidos y de comportamiento

Arquitectura Java Web

- Java nace después de la Web. En la misma estructura del lenguaje se incluyen conceptos y funcionalidad orientada a la web.
- Applets
- Servlets
- Java Platform, Enterprise Edition (Java EE)

Applet

- Componente de una aplicación que se ejecuta en el contexto de otro programa, p.ej. navegador
- No se puede ejecutar de forma independiente
- Realiza una tarea muy específica
- Se ejecuta con privilegios de seguridad muy restringidos

Servlet

- Clase que extiende las capacidades de un servidor, applets que corren en el servidor
- Alternativa Java a programas escritos en PHP o ASP
- Se alojan en contenedores de servlets, como por ejemplo Tomcat de Apache

JavaServer Pages

- Tecnología para el desarrollo de software que crean contenido dinámico para la web, basado en HTML, XML.
- Similar a PHP, pero basado en Java.

Java Platform, Enterprise Edition

- Plataforma de desarrollo en Java para construir aplicaciones distribuidas, en capas, sobre un servidor de aplicaciones
- Actúan de capa intermedia (middleware) para facilitar la interacción entre distintas aplicaciones
- Ejemplo: JBoss de RedHat

Servicios Web

- Tecnología que usa un conjunto de protocolos y estándares para intercambio de datos entre **aplicaciones**.
- Método de comunicación entre dispositivos en la web. Función en ejecución permanente que atiende solicitudes
- Según el W3C: sistema software que permite la interacción de ordenadores a través de una red. Su interfaz se describe mediante un lenguaje específico, p.ej. WSDL.

Representational State Transfer (REST)

- Estilo de arquitectura de software para sistemas distribuidos. Es el modelo de referencia para el diseño de servicios web
- REST describe el comportamiento de los cuatro elementos fundamentales en la web: servidores, pasarelas, proxies y clientes.
- Objetivos:
  - Escalabilidad
  - Generalidad de las interfaces
  - Uso de componentes independientes
  - Uso de componentes intermedios para mejorar latencias, seguridad y abstraer detalles internos

REST: Definición y ventajas

- (BBVAOpen4U): “Cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Alternativa a otros protocolos de intercambio de datos como SOAP.”
- Ventajas:
  - Separación entre el cliente y el servidor
  - Visibilidad, fiabilidad y escalabilidad
  - La API REST siempre es independiente del tipo de plataformas o lenguajes

Escalabilidad de la web, según REST

- Protocolo cliente-servidor sin estado.
- Conjunto de operaciones bien definidas.
- Sintaxis universal para identificar recursos.
- Uso de hipermedios: representados con HTML o XML.

Recursos en REST

- Existen recursos (elementos de información) que pueden ser accedidos usando un URI.
- Para el uso de estos recursos los elementos de la red (clientes y servidores) se comunican mediante una interfaz estándar (HTTP) intercambiando representaciones de los recursos.
- Un aplicación puede interactuar con un recurso conociendo **exclusivamente** su URI y la acción requerida. No necesita conocer de otros elementos intermedios potenciales (proxys, túneles, …).

ACTIVIDADES COMPLEMENTARIAS

Actividades recomendadas

1. Evaluar el diálogo (directo) con un servidor web usando telnet. P.ej.: `telnet betatun.ugr.es 80`
2. Instalar servidores web.
   1. Probar con distintos paquetes (apache, nginx, …)
   2. Estudiar bien las directivas de configuración, particularmente las vinculadas a cuestiones de seguridad
   3. Instalar módulos adicionales (ssl, mysql, php, …)
3. Usar un navegador web con interfaz de texto. P.ej: lynx.
4. Estudiar la estructura y contenidos de los logs de los servidores web

Para profundizar

1. Estudiar el estándar completo del protocolo http.
   Estudiar las diferencias entre http/1.1 y http/2
   Definir los conceptos de conexión, estado y autenticación.
2. Estudiar el funcionamiento de los servidores proxy para web.
   Instalar un servidor de este tipo y configurarlo.
3. Estudiar qué tipo de datos puede recabar un servidor en relación con los usuarios que los utilizan.
4. Estudiar motores de “rendering” de navegadores.