

INTRODUCCIÓN A INTERNET

Diseño Web

Carlos Rojas Sánchez

Licenciatura en Informática

Universidad del Mar



1. Perspectiva histórica de Internet
2. Conceptos relacionados
3. Arquitectura de dos capas y multicapa
4. Protocolo HTTP
5. Evolución del desarrollo de aplicaciones web
6. Tecnologías web
7. Alcances y limitaciones de sitios estáticos

Perspectiva histórica de Internet

Perspectiva histórica de Internet

Internet ha revolucionado la informática y las comunicaciones como ninguna otra cosa. La invención del telégrafo, el teléfono, la radio y el ordenador sentó las bases para esta integración de funcionalidades sin precedentes. Internet es a la vez una herramienta de emisión mundial, un mecanismo para diseminar información y un medio para la colaboración y la interacción entre personas y sus ordenadores, sin tener en cuenta su ubicación geográfica. Internet representa uno de los ejemplos más exitosos de los beneficios de una inversión y un compromiso continuos en el campo de la investigación y el desarrollo de la infraestructura de la información.¹

¹<https://www.internetsociety.org/es/internet/history-internet/brief-history-internet/>

- Intercambiar información
 - Seguridad
 - Privacidad
 - Neutralidad de la red
 - Escalabilidad

Conceptos relacionados

- Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.
- La Corporación de Internet para los Nombres y los Números Asignados (ICANN) es la autoridad que coordina la asignación de identificadores únicos en Internet, incluyendo nombres de dominio, direcciones de Protocolos de Internet, números del puerto del protocolo y de parámetros.
- Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web (WWW o la Web).
- Servicios y protocolos en Internet, ejemplo, www y http.

- Un proveedor de servicios de Internet (o ISP, por la sigla en inglés de Internet Service Provider) es una empresa que brinda conexión a Internet a sus clientes. Un ISP conecta a sus usuarios a Internet a través de diferentes tecnologías como DSL, Cabledem, GSM, Dial-up.

- Internet2 (I2) oUCAID (University Corporation for Advanced Internet Development) es un consorcio sin ánimo de lucro que desarrolla aplicaciones y tecnologías de redes avanzadas, la mayoría para transferir información a alta velocidad.
- Algunas de las tecnologías que han desarrollado han sido IPv6, IP Multicast y Calidad de Servicio (QoS).

- Protocolo de Control de Transmisión (TCP)
- Protocolo de Internet (IP)
- El modelo TCP/IP es un modelo de descripción de protocolos de red desarrollado en los años 70 por Vinton Cerf y Robert E. Kahn.

LA PILA OSI



LA PILA TCP/IP



TCP - Capa de Transporte

- El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.
- También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.
- TCP da soporte a muchas de las aplicaciones más populares de Internet (navegadores, intercambio de ficheros, clientes FTP, etc.) y protocolos de aplicación HTTP, SMTP, SSH y FTP.
- Dado que la capa IP aporta un servicio de datagramas no fiable (sin confirmación), TCP añade las funciones necesarias para prestar un servicio que permita que la comunicación entre dos sistemas se efectúe libre de errores, sin pérdidas y con seguridad.

- IP provee un servicio de datagramas no fiable (también llamado del “mejor esfuerzo”: lo hará lo mejor posible, pero garantizando poco).
- El actual y más popular protocolo de red es IPv4.
- IPv6 es el sucesor propuesto de IPv4; poco a poco Internet está agotando las direcciones disponibles por lo que IPv6 utiliza direcciones de fuente y destino de 128 bits, muchas más direcciones que las que provee IPv4 con 32 bits.

- Direccionamiento
 - Una dirección IP es un número que identifica de manera lógica y jerárquicamente a una interfaz de un dispositivo.
 - Dicho número no se ha de confundir con la dirección MAC que es un número físico que es asignado a la tarjeta o dispositivo de red.
 - DNS y DHCP.
- Enrutamiento.

Uniform Resource Identifier o URI

- URI (en español “identificador uniforme de recursos”) es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.).
- Los URI pueden ser localizadores uniformes de recursos (URL), Uniform Resource Name (URN), o ambos.

Un URI consta de las siguientes partes

- Esquema: nombre que se refiere a una especificación para asignar los identificadores, e.g. urn:, tag:, cid:. En algunos casos también identifica el protocolo de acceso al recurso, por ejemplo http:, mailto:, ftp:.
- Autoridad: elemento jerárquico que identifica la autoridad de nombres (por ejemplo //www.example.com).
- Ruta: Información usualmente organizada en forma jerárquica, que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres (e.g. /domains/example).

Un URI consta de las siguientes partes

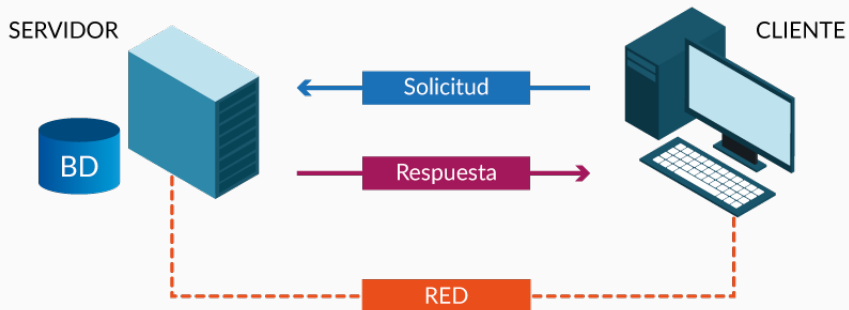
- Consulta: Información con estructura no jerárquica (usualmente pares “clave=valor”) que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres. El comienzo de este componente se indica mediante el carácter ‘?’.
- Fragmento: Permite identificar una parte del recurso principal, o vista de una representación del mismo. El comienzo de este componente se indica mediante el carácter ‘#’.

- La web se desarrolló entre marzo de 1989 y diciembre de 1990 por el inglés Tim Berners-Lee con la ayuda del belga Robert Cailliau mientras trabajaban en el CERN en Ginebra, Suiza, y publicado en 1992.

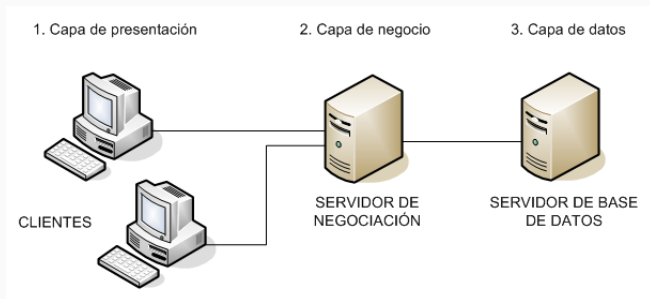
- El Identificador de Recurso Uniforme (URI), que es un sistema universal para referenciar recursos en la Web, como páginas web.
- El Protocolo de Transferencia de Hipertexto (HTTP), que especifica cómo se comunican el navegador y el servidor entre ellos.
- El Lenguaje de Marcado de Hipertexto (HTML), usado para definir la estructura y contenido de documentos de hipertexto.
- El Lenguaje de Marcado Extensible (XML), usado para describir la estructura de los documentos de texto.

Arquitectura de dos capas y multicapa

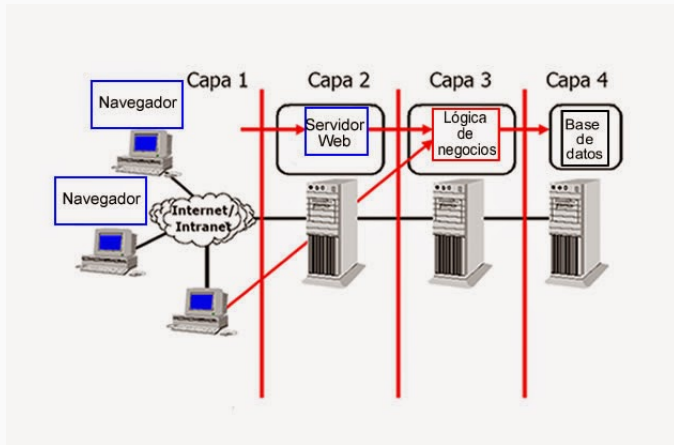
Arquitectura de dos capas



Arquitectura de tres capas



Arquitectura de n-capas



La suposición básica necesaria para poder referirse (referenciar) y hablar de (describir) todos los objetos, es que éstos tengan su nombre propio, que en términos técnicos se llama identificador. En la Web estos nombres propios se llaman Identificadores Universales de Recursos (URI por sus siglas inglesas). Una versión más elemental de URI es la URL (Localizador universal de recursos), que corresponde a una dirección en la Web.

Otra suposición básica para la comunicación universal es un lenguaje único, entendible por todos. Tim BernersLee diseñó el lenguaje HTML (siglas del inglés Hyper Text Markup Language, que a sus características de simplicidad de uso, suma una característica clave: el ser un lenguaje de hipertexto, es decir, que tiene un forma de anclar o redirigir al lector desde un punto cualquiera del texto a otro lugar. Estos son los famosos links o enlaces en la Web.

Desde un punto de vista más técnico, uno necesita un protocolo que permita enviar y traer información en HTML desde un lugar (sitio) a otro en esta gigantesca red que es la Web. El protocolo HTTP (sigla del inglés Hyper Text Transfer Protocol) tiene varias características distintivas que lo han hecho muy perdurable. HTTP es un protocolo de transmisión entre clientes y servidores.

Protocollo HTTP

HTTP en el contexto de la web

El protocolo de transferencia de hipertexto (HyperText Transfer Protocol) es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida.

El protocolo HTTP se basa en un paradigma de peticiones y respuestas.

Los mensajes se pasan con un formato similar al usado por el Internet Mail y el Multipurpose Internet Mail Extensions (MIME).

HTTP en el contexto de la web

Un cliente envía una petición en forma de método, una URI, y una versión de protocolo seguida de los modificadores de la petición de forma parecida a un mensaje MIME, información sobre el cliente y al final un posible contenido. El servidor contesta con una línea de estado que incluye la versión del protocolo y un código que indica éxito o error, seguido de la información del servidor en forma de mensaje MIME y un posible contenido.

El esquema “http” se usa para localizar recursos en la red por medio del protocolo http. La sintaxis de la petición es la siguiente:

“http:” “//” dirección [“:” puerto] [path]

Donde dirección es el nombre de un dominio de Internet o una dirección IP, el puerto es un número que indica el puerto al que se envía la petición y el path indica el recurso al que se accede.

Si no se indica un número de puerto, por defecto se supone que se accede al puerto 80. Si no se indica un path, entonces se supone que este es `"/`.

Los únicos métodos que deben soportar los servidores de forma obligatoria son los métodos GET y HEAD.

²Investigar todos los métodos y para que sirven

El método GET requiere la devolución de información al cliente identificada por la URI. Si la URI se refiere a un proceso que produce información, se devuelve la información y no la fuente del proceso.

El método HEAD es igual que el método GET, salvo que el servidor no tiene que devolver el contenido, sólo las cabeceras. Estas cabeceras que se devuelven en el método HEAD deberían ser las mismas que las que se devolverían si fuese una petición GET.

Métodos POST

El método POST se usa para hacer peticiones en las que el servidor destino acepta el contenido de la petición como un nuevo subordinado del recurso pedido. El método POST se creó para cubrir funciones como la de enviar un mensaje a grupos de usuarios, dar un bloque de datos como resultado de un formulario a un proceso de datos, añadir nuevos datos a una base de datos, ...

- Age, estimación del tiempo transcurrido desde que se creó la respuesta.
- Location, se usa para redirigir la petición a otra URL.
- Proxy-Authenticate, ante una respuesta con el código 407 (autenticación proxy requerida), indica el esquema de autenticación.
- Public, da la lista de métodos soportados por el servidor.

- Retry-After, ante un servicio no disponible da una fecha para volver a intentarlo.
- Server, información sobre el servidor que maneja las peticiones.
- Vary, indica que hay varias respuestas y el servidor ha escogido una.
- Warning, usada para aportar información adicional sobre el estado de la respuesta.
- WWW-Authenticate, indica el esquema de autenticación y los parámetros aplicables a la URI.

Evolución del desarrollo de aplicaciones web

- Sistema basado en hipertexto, que permite clasificar información de diversos tipos , conocido como la gran telaraña mundial
- Funciona por hipertexto y gráficos e incluye y efectos multimedios
- Es considerado como el acceso más sencillo y comprensible al universo de la información disponible en Internet
- Enlaza páginas o documentos localizados en la red sin importar su ubicación física o geográfica

Características de la Web 1.0

- Pocos productores de contenidos.
- Muchos lectores de estos contenidos.
- Páginas estáticas.
- La actualización de los sitios no se realiza en forma periódica.

- Sitios direccionales y no colaborativos.
- Los usuarios son lectores consumidores.
- Interacción mínima reducida a formularios de contacto, inscripción boletines, etc.

- Segunda generación de Web basada en comunidades de usuarios
- Se pasa de una web informativa, creada por expertos a una web social, donde cualquiera puede participar fácilmente
- Aparecen aplicaciones Web muy potentes y sencillas de manejar enfocadas al usuario final
- Basa su desarrollo en CMS (Sistemas de Gestión de Contenidos)

¿Qué es un CMS?

- Un Sistema de Gestión de Contenidos (Content Management System), permite la creación y administración de contenidos principalmente en páginas web
- Consiste en una interfaz que controla una o varias bases de datos donde aloja el contenido
- El sistema permite manejar de manera independiente el contenido y el diseño

Evolución de los CMS

- Páginas estáticas (HTML). Edición a mano. Difícil actualización. Contenido y diseño unidos.
- Páginas dinámicas (CGI). Gestores complicados. Poca flexibilidad.
- Páginas dinámicas (PHP, ASP, Java). Gran flexibilidad. Crecimiento de las comunicaciones de usuario. Separación total entre presentación y contenido.

El usuario es el centro Protagonista usuario que crea y comparte

Participación Conocimiento compartido en base a la suma de esfuerzos individuales. Cuantos más usuarios comparten mayor utilidad del servicio

Usabilidad (facilidad de utilización de las herramientas) No es necesario grandes conocimientos técnicos para crear un espacio en Internet con contenidos que sean accedidos por otros usuarios

AJAX, acrónimo de Asynchronous JavaScript And XML
(JavaScript asíncrono y XML)

No es ninguna tecnología, ni lenguaje de programación

Es una técnica de desarrollo web que combina varias tecnologías

Consiguiendo una navegación más ágil y rápida, más dinámica

Tecnologías empleadas

XHTML (o HTML) y CSS Presentación de datos

Document Object Model (DOM) Mostrar e interactuar
dinámicamente con la información

XML y XSLT Intercambiar y manipular datos con el servidor web

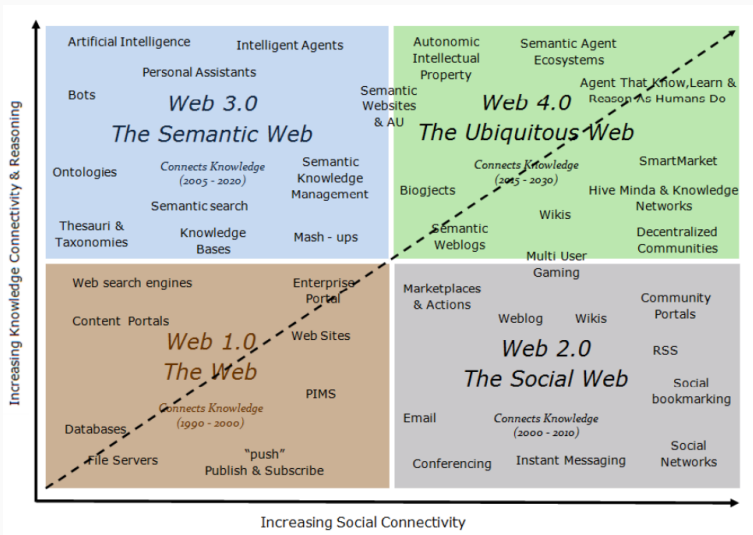
XMLHttpRequest Recuperación y envío de datos de modo
asíncrono

JavaScript Nexos de unión

- La Web Semántica es una extensión de la Web en la cual la información se da mediante un significado bien definido, lo que facilita que los ordenadores y la gente trabajen en cooperación
- No se enlazan con la totalidad de páginas existentes
- La escasa precisión de los resultados
- La alta sensibilidad al vocabulario empleado en la búsqueda

- Conexión Inteligente
- Algoritmos de procesamiento de lenguaje natural
- Desarrollo de Agentes Inteligentes
- Desarrollo de Comunidades Semánticas

Evolución del desarrollo de aplicaciones web



Tecnologías web

Programación web: lenguajes del lado servidor y del lado cliente

- La web funciona según un principio realmente sencillo: los servidores web albergan los contenidos y los clientes los solicitan por HTTP o FTP
- Clientes - Navegadores (Mozilla Firefox o Google Chrome)
- Servidores - Los servidores web, como Apache o NGINX, forman parte de los proyectos de desarrollo web, se instalan y se ejecutan en este entorno y permiten a los clientes el acceso a los contenidos.

Programación del lado del servidor

- Esta tecnología de desarrollo web se basa en la utilización de scripts que ejecuta el servidor web, con ayuda del lenguaje de programación adecuado, cuando un cliente solicita el contenido.
- Una tarea frecuente de los scripts consiste en extraer los datos que se necesitan de una base de datos y de integrarlos en el proyecto web. Aunque el usuario accede al proyecto a través de páginas HTML, el código fuente de los scripts permanece oculto.

- El uso de estos scripts del lado del servidor presupone que el cliente seguirá enviando peticiones al servidor web para entregar información nueva y modificada al usuario.
- Esto, por un lado, conlleva una fuerte carga de la capacidad del servidor, con la consecuente influencia en su tiempo de respuesta y, por otro, hace imprescindible una conexión al servidor para poder acceder a la página.

Programación del lado del servidor

	ASP.NET	Java	Ruby
Desarrollado por	Microsoft	Sun Microsystems	Yukihiro Matsumoto et al.
Licencia	propietario	GNU GPL	BSD
Año de publicación	2002	1995	1995
Plataforma	Windows	independiente	independiente
Paradigmas de programación	Orientado a objetos	Orientado a objetos	multiparadigma

Programación del lado del servidor

	Perl	PHP	Python
Desarrollado por	Larry Wall, et al.	Rasmus Lerdorf	Guido van Rossum, Python Software Foundation
Licencia	GNU GPL y licencia artística	Licencia PHP, entre otras	Python Software Foundation Licence
Año de publicación	1987	1995	1991
Plataforma	Independiente	Independiente	Independiente
Paradigmas de programación	procedural, modular, en parte orientado a objetos	imperativo, funcional, orientado a objetos	multiparadigma

Programación del lado del cliente

- Los lenguajes de programación del lado cliente también se utilizan para realizar proyectos con contenidos dinámicos, pero, a diferencia de los lenguajes del lado servidor, no es el servidor el que ejecuta y procesa los scripts, sino el cliente solicitante.
- Cuando un usuario solicita una página web o una aplicación web con un script de este tipo, el servidor web envía el documento HTML y el script al navegador, quien lo ejecuta y presenta el resultado final.

- El lenguaje del lado cliente más significativo es JavaScript, desarrollado por Brendan Eich de Netscape (Netscape Communications Corporation), empresa de software responsable de la creación del navegador predecesor de Mozilla Netscape, y publicado en 1995 junto con la versión anterior del navegador Navigator 2.0, aún con el nombre de LiveScript.

Alcances y limitaciones de sitios estáticos

Página web estática es aquella que muestra el mismo contenido a todos los usuarios, sin cambios en función de la interacción del usuario. Los archivos son precontruidos compuestos por HTML, CSS y JavaScript.

Un sitio web estático no tiene procesamiento backend del lado del servidor ni base de datos. Cualquier funcionalidad dinámica asociada al sitio web estático se realiza en el lado del cliente, lo que significa que el código se ejecuta en los navegadores de los visitantes y no en el servidor.

Alcances de los sitios estáticos

- Más fáciles de optimizar en cuanto a rendimiento
- Requieren menos recursos del servidor
- Alojamiento de sitios web estáticos es muy barato
- Más fácil de proteger
- Muy fácil de lanzar sitios web sencillos

Limitaciones de los sitios estáticos

- Puede ser más complejo aplicar actualizaciones de contenido/diseño
- Puede ser más técnico añadir funciones
- La funcionalidad de gestión de contenidos no suele ser tan potente
- Dependen de servicios de terceros incluso para funcionalidades básicas como los formularios web

Trabajo Parcial 1

Crear tu web personal con Github.io

- Prerequisito tener cuenta en Github
- Seguir los pasos siguientes:



- Template

