

# Introducción: Implantación de arquitecturas web

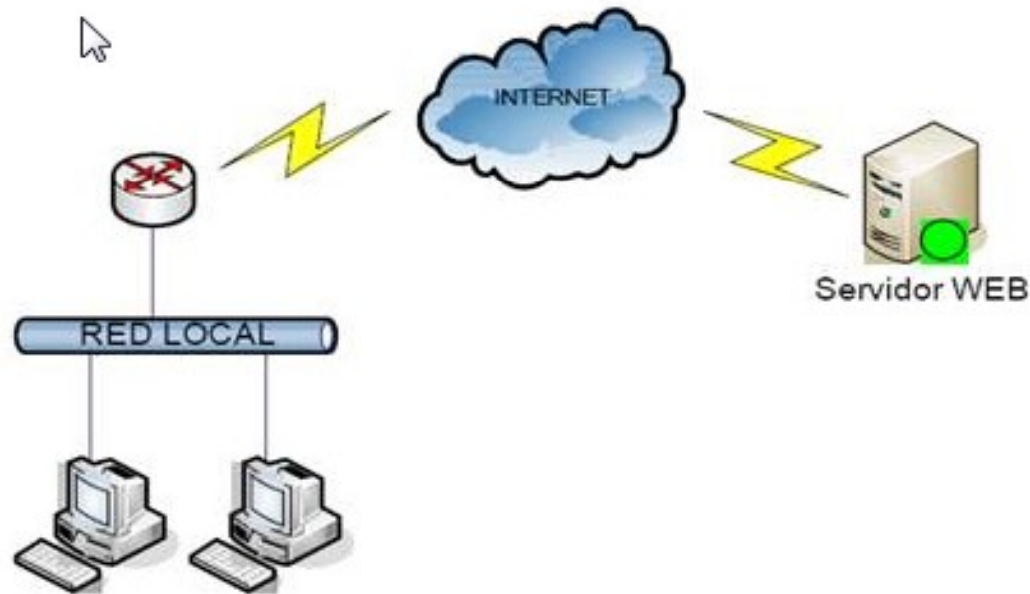


Diagrama representativo de un **Servidor web**

# Internet

- Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de **protocolos TCP/IP**.
- 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

# ¿Qué es la WWW?

- Uno de los servicios que más éxito ha tenido en Internet ha sido la **World Wide Web (WWW)** “Tela de Araña Mundial” o “**la Web**”. Es una red de recursos de información, estructurada en forma de hipertexto y que funciona sobre la red **Internet**.
- La web (1990) es un conjunto de protocolos, estándares y tecnologías, basadas en Internet, diseñado originalmente para la consulta remota de información en archivos de hipertexto.
- Con un **navegador web**, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces. Las páginas web se descargan desde un servidor web.

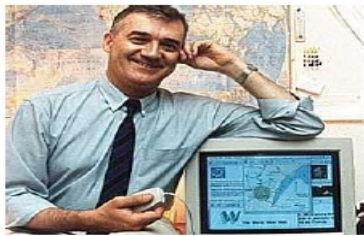
# Otros servicios en Internet

Existen muchos otros servicios y protocolos en Internet:

- Envío y recepción de correo electrónico (POP3, IMAP, SMTP)
- Transmisión de archivos (SSH, FTP, P2P, HTTP)
- Mensajería instantánea (IRC, XMPP)
- Transmisión de contenido multimedia (VoIP, RTP)
- Conexión remota por consola (SSH, Telnet)
- Conexión remota gráfica (VNC)
- CONSULTA=> <http://es.wikipedia.org/wiki/Internet>

# La Web (WWW)

- La Web fue creada alrededor de 1989 por Tim Berners-Lee y Robert Cailliau mientras trabajaban en el CERN
- Se publicó en 1992



Robert Cailliau



Tim Berners-Lee

- 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 el **Hiper Text Transfer Protocol (HTTP)**
- Las páginas se identifican con un nombre único llamado coloquialmente **dirección web (URL)**

# Página web

- Una página web es un **documento electrónico** adaptado a la web.
- Está diseñada para ser visualizado en un **navegador web**, que la descargará de un **servidor web**.
- Contiene **texto, gráficos y contenido multimedia**
- Contiene hiperenlaces integrados en el texto que permiten al usuario navegar o acceder a diferentes páginas web relacionadas entre sí.
- Una página web generalmente es un fichero con extensión .html o .xhtml . Está escrita con formato de marcado textual **HTML o XHTML**
- Puede contener información de estilos (colores, tipos de letra, distribución de elementos,...) en el formato **CSS** integrado en el propio documento o en un fichero diferente.
- Puede enlazar a imágenes en formato **JPG o PNG**

# 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**)
- Ejemplos:
  - =><http://www.urjc.es>: Sitio web de la URJC
  - =><http://www.agenciatributaria.es> Sitio web de la agencia tributaria
- Tipos de sitios web: Web corporativas de empresas, Blogs, redes sociales, Páginas personales, Buscadores, Sitios de noticias, Enciclopedias, ...
- También se usa en ocasiones en término **Portal** para referirse a un sitio web o **WebSite**.

# Aplicación Web

- En los comienzos de la web, 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 en vez de navegación secuencial. La edición de sitios web se realizaba con herramientas similares a la edición de documentos (p.e. Microsoft FrontPage). A estas páginas se las denominaba **páginas web estáticas**.
- Poco a poco las páginas empezaron a ser más **dinámicas**, en vez de ser ficheros .html en el disco, empezaban a ser pequeños programas que se ejecutaban cada vez que un usuario pedía una página. Inicialmente eran cambios mínimos (contador de visitas, fecha actual, cambiar la imagen de cabecera...) con lenguajes de script como **PERL y PHP (principios de la década de los 90)**
- Pero pronto los cambios serían cada vez más profundos y las páginas web se convertirían en completas **aplicaciones web** (Software que se codifica en un lenguaje soportado por los navegadores web).



# Aplicación Web

Una **aplicación web** es aquella aplicación que los usuarios pueden utilizar accediendo a un servidor web a través de Internet mediante un navegador .

- Es una aplicación software que se implementa para que su interfaz de usuario se utilice desde un navegador web.
- **Características:**
  - ✓ Utilizan 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 potenciales
  - ✓ Comunicación activa entre el usuario y la información
  - ✓ Utilización de lenguajes interpretados en el lado del cliente, o de plugins tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario.

# Ejemplos de Aplicaciones Web

- **Webmails**
- **Wikis**
- **Weblogs**
- **Tiendas en línea**
- **Wikipedia**

Actualmente, en la práctica, todas las webs del mundo son aplicaciones web porque la mayoría tienen cierto contenido dinámico.

# Estructura de las aplicaciones web

- El **navegador web** ofrece la primera capa
- Un **motor** capaz de usar alguna tecnología web dinámica (ejemplo: PHP, Java Servlets o ASP, ASP.NET, CGI, Ajax, ColdFusion, embPerl, Python (programming language) o Ruby on Rails) constituye la capa de en medio.
- Por último, una **base de datos** constituye la tercera y última capa.

# **Características de las aplicaciones web**

- **Ahorra tiempo**
- **No hay problemas de compatibilidad**
- **No ocupan espacio** en nuestro disco duro.
- **Actualizaciones inmediatas**
- **Bajo consumo de recursos**
- **Multiplataforma**
- **Portables**
- **La disponibilidad suele ser alta**
- **Los virus no dañan los datos**
- **Colaboración**

# PROBLEMAS CON LAS APLICACIONES DE ESCRITORIO

- Duplicidad de datos por la falta de unificación de los mismos.
- Diseminación de la información y lógica en muchas partes (cada computador que la use).
- Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- Problemas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- La administración de la seguridad, controlando el acceso a los usuarios a información no relevante o privada es un caos.
- Dificultad para configurar cada una de las instalaciones (*deployments*) dependiendo de las necesidades de cada usuario.

# VENTAJAS DE LAS APLICACIONES WEB FRENTE A LAS APLICACIONES DE ESCRITORIO

- No requiere instalar software especial (en los clientes).
- Bajo coste en actualizar los equipos con una nueva versión.
- Acceso a la última y mejor versión.
- Información centralizada.
- Seguridad y copias de seguridad.
- Movilidad.
- Reducción de costes en los puestos cliente (mayor longevidad).

# Evolución Histórica de La WEB

- Web 1.0:

De 1993 – 2000

- ✓ Páginas con hiperenlaces
- ✓ El usuario sólo leía contenido publicado por otros
- ✓ Contenido muy estático, muy difícil de editar
- ✓ El navegador web era la única aplicación “conectada”

A partir de 2000

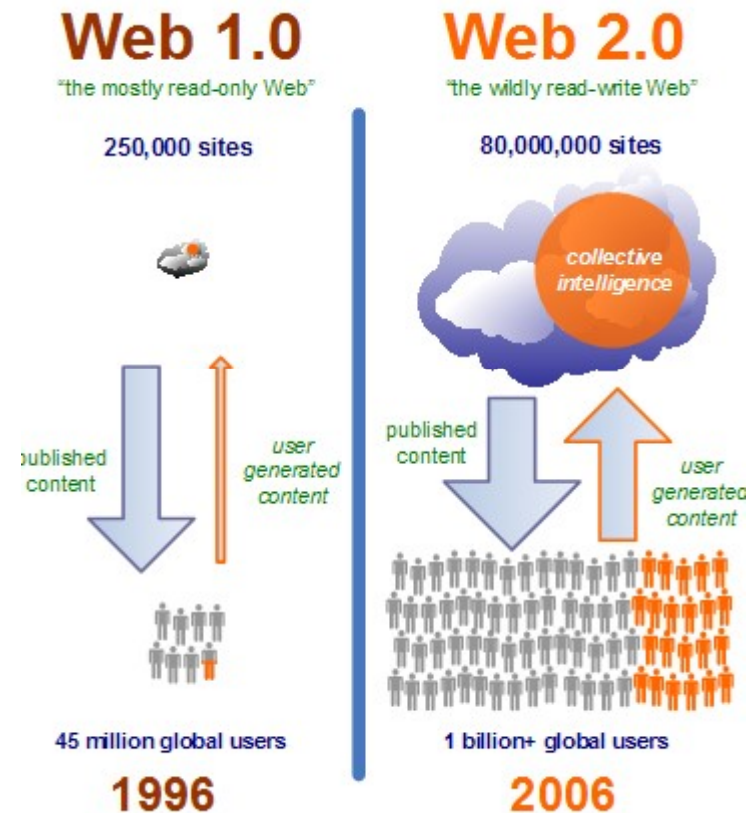
- ✓ Las tecnologías se desarrollan
- ✓ Los usuarios tienen más facilidades para editar el contenido de las páginas web...
- ✓ Nacen los blogs, los wikis...
- ✓ Google se empieza a popularizar
- ✓ Nace la wikipedia (2001)



# Web 2.0

Los usuarios toman el control de los contenidos

- Web 1.0: Web de Lectura Individual
- Web 2.0: Web de Lectura / Escritura Social





# Web 2.0

- El término Web 2.0 fue acuñado por Dale Dougherty (O'Reilly) en 2004
- Web como plataforma con aplicaciones ligeras, dinámicas y en constante evolución
- Incrementa la colaboración en la creación y distribución de contenidos.
- Múltiples dispositivos de acceso

## Componentes de la Web 2.0:

- ✓ Aplicaciones Ricas de Internet (RIA)
- ✓ Arquitectura Orientada al Servicio (SOA)
- ✓ Web Social



# Aplicaciones de Internet Enriquecidas

## *Rich Internet Applications (RIA)*



- Son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales.
- Utilizan un navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.
- No producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una base de datos o de otros fichero.
- Ejemplos: Google Maps, Facebook, Office 365, Google Docs.

# Arquitectura Orientada al Servicio (SOA)

*Service Oriented Architecture*



- Es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos, aglutina el conjunto de tecnologías y técnicas que permiten diseñar aplicaciones como un conjunto de servicios que resuelven peticiones de los usuarios. De esta forma se crean elementos software reutilizables que son independientes del lenguaje con el que fueron creados.

# Web social



- La Web social está integrada por la llamada comunidad virtual dentro de la que podemos encontrar interacción con redes de ordenadores con servicios tales como p2p, correo electrónico, redes sociales, blogs, podcast, webcast, wikis, marcadores sociales, sindicación de contenidos RSS, imágenes, video digital hospedado en servidores de video y audio digital.
- El uso de este entorno ha democratizado el uso de servicios y aplicaciones ofreciendo a todos los usuarios la posibilidad de crearlos, utilizarlos, compartirlos y distribuirlos; es decir es el punto de encuentro en la web social.

# Web 3.0



- Web 3.0 es la web que facilita la accesibilidad de las personas a la información, sin depender de qué dispositivo use para el acceso a ella diseñada bajo parámetros de rendimiento eficiente, optimizando los tiempos de respuesta, optimizando los consumos energéticos globales del sistema, enfocada a ser herramienta para el desarrollo sostenible.
- Está relacionada con la inteligencia artificial, la web semántica, la Web Geoespacial o la Web 3D.
- **Web Inteligente:** Inteligencia artificial aplicada a la web, se refiere a que la propia web reaccione de forma inteligente ante los usuarios.
- **Web semántica:** Se trata de que los contenidos en la web dispongan de etiquetados o metadatos que permitan darles significados.

# Navegadores y servidores web

- La web sigue una arquitectura **cliente-servidor**.
- La **arquitectura cliente-servidor** es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**. Un cliente realiza peticiones a otro programa, el **servidor** que le da respuesta.
- El **navegador web** actúa como cliente
- La comunicación se establece usando el protocolo **http**
- Un **navegador web** es una aplicación que se instala en el sistema que utiliza el usuario
- El usuario escribe una dirección web (**URL - Uniform Resource Locator**). *La dirección contiene el nombre del servidor web y el nombre del recurso que se solicita.*
- El navegador hace una petición al servidor y solicita el recurso
- El navegador descarga el recurso y lo visualiza
- Si el recurso es una página HTML, además de visualizar su contenido, descarga recursos adicionales como imágenes, estilos, etc. y los visualiza integrados en la página.

# Arquitectura cliente servidor.

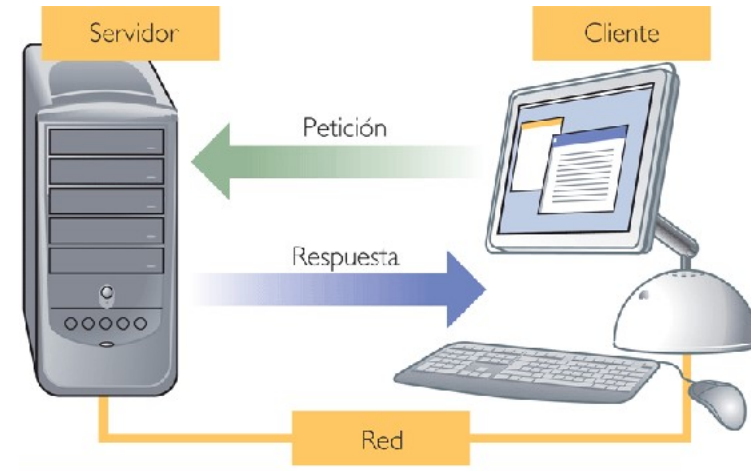
El **cliente** es el **remitente de una solicitud**.

Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.

Espera y recibe las respuestas del servidor.

Por lo general, puede conectarse a varios servidores a la vez.

Normalmente interactúa directamente con los usuarios finales mediante una **interfaz gráfica de usuario**.

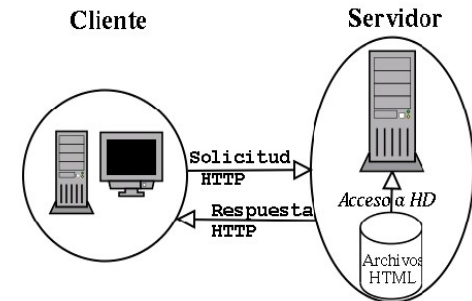


El **Servidor** es el **receptor de la solicitud** enviada por el cliente

Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación.
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No suelen interactuar directamente con los usuarios finales.

# Arquitectura cliente servidor



- Un **servidor web** o **servidor HTTP** es un programa informático que procesa una aplicación del lado del servidor. Es fundamentalmente un programa que escucha en un puerto a la espera de conexiones.
- Una petición HTTP identifica el recurso que le interesa al cliente y le dice al servidor qué "acción" realizar con el recurso.
- Es un protocolo **sin estado**, no guarda ninguna información sobre las peticiones de páginas web realizadas anteriormente.



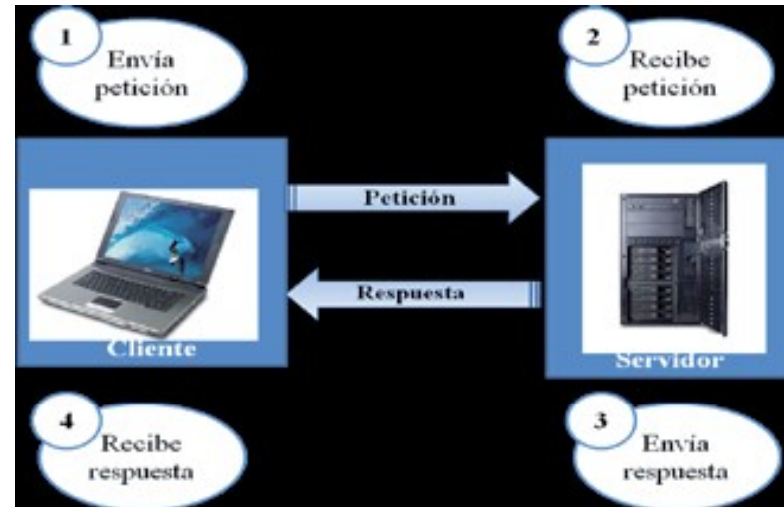
# Características Arquitectura Cliente/Servidor:

## Ventajas:

- Centralización del control.
- Escalabilidad.
- Fácil mantenimiento.
- Tecnologías.

## Desventajas:

- Congestión del tráfico
- Cuando un servidor está caído las peticiones de los clientes no pueden ser satisfechas
- El software y el hardware de un servidor son generalmente muy determinantes
- El cliente no dispone de los recursos que puedan existir en el servidor.



# Cliente web

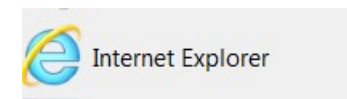
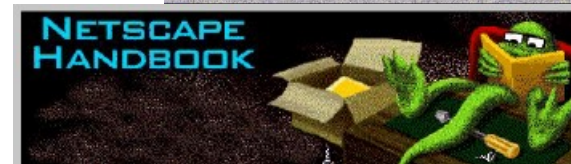
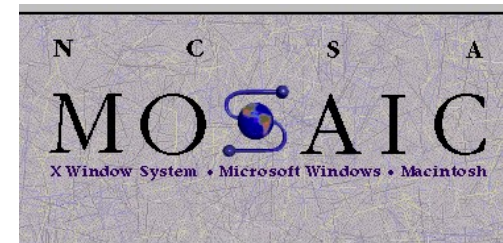
- El cliente web por excelencia es el **navegador web**.
- Existen un conjunto de **estándares web**, definidos por el **W3C**, que todo navegador debería implementar.
- Existen un conjunto de tecnologías no estándar que algunos navegadores implementan para la construcción de aplicaciones avanzadas y acceso a contenido multimedia .
- El W3C (*World Wide Web Consortium*) es una comunidad internacional que desarrolla estándares abiertos que aseguran el crecimiento de la Web a largo plazo.

<http://www.w3.org>



# Navegadores web

- 1990 **WorldWideWeb**
  - Nace el primer navegador para sistemas NeXT
- 1993 **Mosaic**
  - Win, Mac y Unix/Linux.
  - Fue la base de Firefox e Internet Explorer
  - Cuota de mercado de 90% en 1994
- 1994 **Netscape Navigator**
  - Como evolución de Mosaic
- 1995 **Microsoft Internet Explorer**
  - Integrado en Windows
  - Llegó a tener el 95% de cuota en el 2002
- 1996 **Opera**
  - Nunca ha tenido una gran cuota de mercado
  - Actualmente se utiliza en móviles y consolas (Nintendo Wii)



# Navegadores web

- 2003 **Apple Safari**



- Navegador de productos de Apple
- Basado en motor de renderizado WebKit (libre)

- 2004 **Mozilla Firefox**



- Software libre
- Usa el motor Gecko
- Desarrollado por la fundación Mozilla (heredera de Netscape)

- 2008 **Google Chrome**



- Tiene una versión software libre llamada Chromium
- Basado en motor de renderizado WebKit (libre)

# Servidores Web

Un servidor web se compone:

## 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 web (usando http)

## Hardware

- Servidor en el que se ejecuta el software de servidor web.
- Generalmente dispone de otros servicios adicionales como un sistema gestor de base de datos
- 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 **URL** que contiene: Nombre del servidor, Ruta del fichero en el disco y Nombre del fichero
- Ejemplo:  
`http://www.miservidor.com/ruta/del/fichero/fichero.txt`

# Servidores Web

- La mayoría de los servidores web permiten que en cada petición se ejecute un pequeño 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 con contenido dinámico no sólo envían recursos al navegador, también pueden procesar información que les llega del mismo:
    - ✓ Datos contenidos en formularios web
    - ✓ Ficheros enviados desde el navegador
    - ✓ Información codificada en los links que pulsan los usuarios (URL)
- => Esta funcionalidad permite el desarrollo de completas **aplicaciones web**

# Servidores Web

- 1990 - CERN http (NeXTSTEP)

**WorldWideWeb** es un navegador web y editor de páginas web en modo gráfico para el sistema operativo NEXTSTEP; es reconocido como la primera aplicación informática de este tipo. Tiempo después de su lanzamiento fue renombrado **Nexus** para evitar confusiones con el concepto de Word Wide Web

## WorldWideWeb



Navegador WorldWideWeb

**Desarrollador**

**Tim Berners-Lee** para el CERN  
[w3.org/People/Berners-Lee/WorldWideWeb.html](http://w3.org/People/Berners-Lee/WorldWideWeb.html)

**Información general**

Lanzamiento inicial	23 de diciembre de 1990
Última versión estable	0.18  14 de enero de 1994; hace 21 años
Género	Navegador web
Sistema operativo	NEXTSTEP
Licencia	Dominio público
En español	No

[\[editar datos en Wikidata\]](#)

# Servidores Web



- 1995 – **Apache**
- ✓ Su nombre completo es **Apache HTTP Server Project**
- ✓ El servidor más popular (Enero 2014 – 42% cuota)
- ✓ Desarrollado por la Apache Software Foundation con licencia software libre Apache License
- ✓ Multiplataforma
- ✓ Permite escribir **contenido dinámico** (*server-side scripting*)

<http://httpd.apache.org>

## Apache HTTP Server

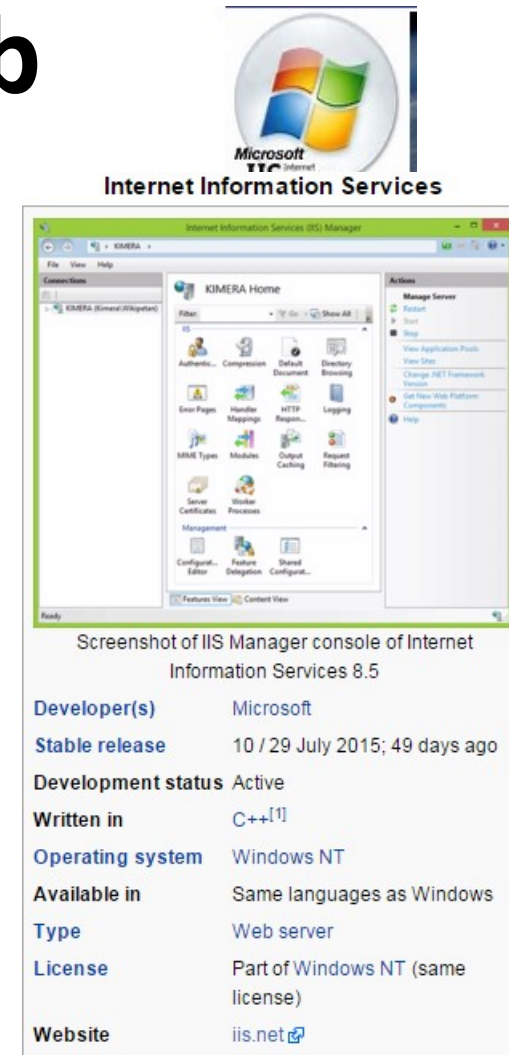


Original author(s)	Robert McCool
Developer(s)	Apache Software Foundation
Initial release	1995; 20 years ago <sup>[1]</sup>
Stable release	2.4.16 (July 15, 2015; 2 months ago) [±]
Development status	Active
Written in	C/C++, <sup>[2]</sup> XML <sup>[3]</sup>
Operating system	Windows, OS X, GNU/Linux, Unix, FreeBSD, Solaris, NetWare, OS/2, TPF, OpenVMS and eComStation
Type	Web server
License	Apache License 2.0
Website	<a href="http://httpd.apache.org">http://httpd.apache.org</a>



# Servidores Web

- 1995 – Microsoft Internet Information Services (IIS)
- ✓ Es un producto propietario de Microsoft integrado en su sistema operativo Windows y su base de datos **MS SQL Server**
- ✓ Permite escribir **contenido dinámico (*server-side scripting*)**
- ✓ Es el segundo servidor más usado a (Enero 2014 con 30%)




# Servidores Web

- 2004 – **NginX**
  - ✓ Es un servidor multiplataforma y software libre (licencia tipo BSD)
  - ✓ Desarrollado por **Igor Sysoev**
  - ✓ Es el tercer servidor más usado a (Enero de 2014 con 14%)
  - ✓ Permite escribir **contenido dinámico (*server-side scripting*)**
  - ✓ Es muy rápido y se usa como balanceador de carga
  - ✓ El sistema es usado en muchos sitios web, como: WordPress, Netflix, Hulu, SourceForge, TorrentReactor y partes de Facebook (como el servidor de descarga de archivos zip pesados)
- => <http://nginx.net/>

nginx	
<b>NGINX</b>	
Desarrollador	
Igor Sysoev	
<a href="http://nginx.org/">http://nginx.org/</a>	
Información general	
Lanzamiento inicial	04 de octubre de 2004
Última versión estable	1.8.0 (info) 21 de abril de 2015; hace 4 meses
Última versión en pruebas	1.9.0 (info) 28 de abril de 2015; hace 4 meses
Género	Servidor web, Proxy
Programado en	C
Sistema operativo	Multiplataforma
Licencia	Licencia BSD simplificada. <sup>1</sup>
En español	No
<small>[editar datos en Wikidata]</small>	

# Servidores Web

**Cherokee** es un servidor web multiplataforma. Su objetivo es ser rápido y completamente funcional, sin dejar de ser liviano comparado con otros servidores web. Está escrito completamente en C. Puede usarse como un sistema embebido y soporta complementos para aumentar sus funcionalidades. Es software libre, disponible bajo la Licencia Pública General de GNU.

Cherokee	
	
Desarrollador	
Cherokee Project <a href="http://www.cherokee-project.com/">http://www.cherokee-project.com/</a>	
Información general	
Diseñador	Álvaro López Ortega y la comunidad Cherokee
Última versión estable	1.2.101 ( <a href="#">info</a> ) 18 de octubre de 2011; hace 3 años
Género	Servidor web
Programado en	C
Sistema operativo	Multiplataforma
Licencia	Licencia Pública General de GNU
Idiomas	Multilingüe <sup>1</sup>
En español	Sí
<a href="#">[editar datos en Wikidata]</a>	

# Otros Servidores Web

- **Oracle iPlanet Web Server 7.0** es un servidor web de alto rendimiento, ampliamente escalable y seguro que ofrece un contenido dinámico y estático. Su virtualización de dominios, la versatilidad de su configuración y su robusta seguridad le permiten ofrecer un servicio con una mejor calidad
- **Lighttpd** es un servidor web diseñado para ser rápido, seguro, flexible, y fiel a los estándares. Está optimizado para entornos donde la velocidad es muy importante, y por eso consume menos CPU y memoria RAM que otros servidores. Es software libre y se distribuye bajo la licencia BSD. Funciona en GNU/Linux y UNIX de forma oficial. Para Microsoft Windows actualmente hay una distribución.
- Servidores web **incrustados**

El interfaz web se ha convertido en el interfaz universal para las aplicaciones distribuidas

Muchos dispositivos incluyen un servidor web incrustado (*embeded*) para *gestionar y configurar su comportamiento*

Ejemplos : Routers, Impresoras, Cámaras IP, Teléfonos móviles...

# Programación del lado del servidor/back-end

- *Server-side scripting* se puede traducir por “Programación del lado del servidor” .
- La tecnología que conectaba el servidor web con un programa para generar las páginas web se denominaba **CGI (Common Gateway Interface)**
- Los programas, llamados **CGIs**, se ejecutaban completamente en cada petición http que se hacía al servidor. Estos programas estaban implementados en lenguajes como **C o scripts de Perl, Shell y PHP**
- Debido a los **problemas de rendimiento** con los CGIs, surgieron nuevos mecanismos para la programación del lado del servidor.
- Se desarrollan técnicas para incluir las aplicaciones dentro de los servidores (módulos de Apache, etc...)
- Existen muchas tecnologías para la programación del lado del servidor. Las más conocidas son **Java EE, PHP, ASP.NET, Ruby on Rails, Django, Node.js...**

# Programación del lado del servidor

## Java Enterprise Edition (Java EE)



- Tecnología basada en **Java**
- Desarrollada por una coalición de empresas lideradas por **Oracle, IBM, Red Hat, etc..**
- Tecnología muy usada a nivel **empresarial**
- La mayoría de las implementaciones y herramientas para desarrollo son **software libre**
- Las aplicaciones se ejecutan en servidores web implementados en Java (**Tomcat, Glassfish, JBoss, Jetty...**)
- Estos servidores se integran en los servidores web **Apache, NginX e IIS**

<http://www.oracle.com/javaee/>

# Programación del lado del servidor



## PHP

- Desarrollado en 1994 por **Rasmus Lerdorf**
- Con licencia libre PHP license
- Es una tecnología multiplataforma que se integra bien con servidores como **Apache, NginX e IIS**
- Se puede usar **Netbeans o Eclipse** para su desarrollo
- **Facebook** está implementado con PHP

<http://www.php.net/>

# Programación del lado del servidor



## ASP.NET

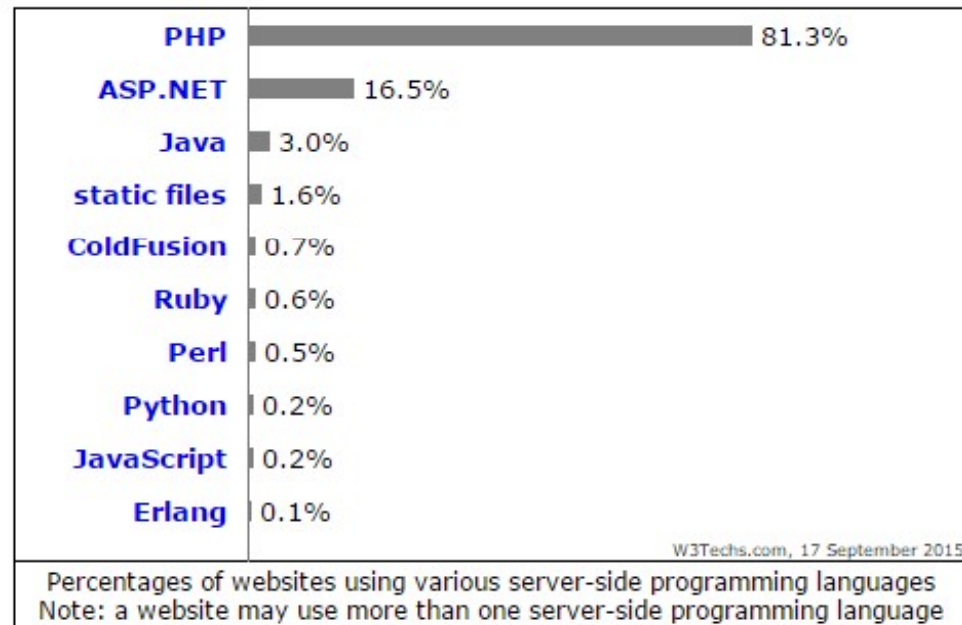
Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

- Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP).
- Forma parte de la **tecnología .NET de Microsoft**
- Se utiliza el lenguaje **C#**
- Licencia **propietaria** y para plataformas Windows
- Se integra bien con el servidor **IIS**
- El desarrollo se realiza con **Visual Studio .NET**

<http://www.asp.net/>



# Programación del lado del servidor



[http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all)

# Tecnologías de desarrollo de aplicaciones web

- Bases de datos:



- Las bases de datos más populares para el desarrollo de aplicaciones web han sido las **bases de datos relacionales**

- Existen muchas bases de datos relacionales (comerciales y software libre): MySQL, Derby, Oracle, MS SQL Server, PostgreSQL



# Bases de Datos Relacionales



## MySQL

- <http://www.mysql.org/>
- Sistema gestor de base de datos multiplataforma
- Desarrollado en C
- Licencia código abierto GPL
- Soporte de un subconjunto de SQL 99
- Herramienta interactiva para hacer consultas y crear bases de datos
- Muy popular en el desarrollo web

# Bases de Datos NoSql

- Como las aplicaciones web tienen muchas necesidades de escalabilidad y tolerancia a fallos, hay una **nueva familia de bases de datos**
- Se denominan genéricamente **NoSQL**, que se puede interpretar **como: No SQL o como Not Only SQL**
- Algunas de las más famosas son:



# Sistemas gestores de contenido

- **CMS** (*Content Management System*)
- Aplicación web genérica que permite la creación y administración de contenidos **vía web**
- El sistema permite manejar de manera independiente el contenido y el diseño, permite el cambio de diseño (con *templates o themes*)
- Los CMSs han evolucionado para convertirse en un **nuevo modelo de desarrollo de aplicaciones web configurando y adaptando módulos con un interfaz web**

# CMS

- Existen multitud de CMSs con enfoques y objetivos diferentes
- Ejemplos: Drupal (PHP), Joomla (PHP), Wordpress (PHP), Wikimedia(PHP), Moodle (PHP), Liferay (Java)



# HTTP: Hypertext Transfer Protocol

- El hipertexto es el contenido de las páginas web y está escrito según indica el lenguaje HTML.
- El protocolo de transferencia HTTP es el sistema mediante el cual se envían las peticiones para acceder a una página web, y se reciben las respuestas del servidor web (las páginas web). HTTP también permite enviar datos al servidor para interactuar con él, como por ejemplo formularios de datos.

# Características de HTTP

Protocolo *Hypertext Transfer Protocol estándar (W3C y IETF)*

- La versión 1.0 se publicó en **1996**
- Los paquetes están codificados en **texto plano** y con un formato sencillo (no en binario)
- 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**, es decir, que no guarda ninguna información sobre conexiones anteriores



# 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**
- Enviará la petición con formato :

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

# Ciclo de vida de una petición http

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)
.
.
.
</body>
</html>
```

# 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

apache.pcap [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter:  Expression... Clear Apply

Time	Source	Destination	Protocol	Length	Info	Src MAC
0.000000	192.168.1.132	192.168.1.135	TCP	78	49182 > http [SYN] Seq=0 win=65535 Len=0 MSS=1460	Micro-
0.000009	192.168.1.135	192.168.1.132	TCP	74	http > 49182 [SYN, ACK] Seq=0 Ack=1 win=5792 Len=0	Apple_
0.000717	192.168.1.132	192.168.1.135	TCP	66	49182 > http [ACK] Seq=1 Ack=1 win=524280 Len=0	Micro-
0.001134	192.168.1.132	192.168.1.135	HTTP	426	GET / HTTP/1.1	Micro-
0.001160	192.168.1.135	192.168.1.132	TCP	66	http > 49182 [ACK] Seq=1 Ack=361 win=6912 Len=0	Apple_
0.003422	192.168.1.135	192.168.1.132	HTTP	419	HTTP/1.1 200 OK (text/html)	Apple_
0.004618	192.168.1.132	192.168.1.135	TCP	66	49182 > http [ACK] Seq=361 Ack=354 win=524280 Len=0	Micro-
5.003424	192.168.1.135	192.168.1.132	TCP	66	http > 49182 [FIN, ACK] Seq=354 Ack=361 win=6912 Len=0	Apple_
5.111552	192.168.1.132	192.168.1.135	TCP	66	49182 > http [ACK] Seq=361 Ack=355 win=524280 Len=0	Micro-
29.864048	192.168.1.132	192.168.1.135	TCP	66	49182 > http [FIN, ACK] Seq=361 Ack=355 win=524280 Len=0	Micro-
29.864082	192.168.1.135	192.168.1.132	TCP	66	http > 49182 [ACK] Seq=355 Ack=362 win=6912 Len=0	Apple_

Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)

Ethernet II, Src: Apple\_1b:c1:9c (f0:b4:79:1b:c1:9c), Dst: Micro-St\_ad:4c:15 (00:13:d3:ad:4c:15)

Internet Protocol Version 4, Src: 192.168.1.132 (192.168.1.132), Dst: 192.168.1.135 (192.168.1.135)

Transmission Control Protocol, Src Port: 49182 (49182), Dst Port: http (80), Seq: 0, Len: 0

Source port: 49182 (49182)

Destination port: http (80)

[Stream index: 0]

Sequence number: 0 (relative sequence number)

Header length: 44 bytes

Flags: 0x02 (SYN)

Window size value: 65535

# Aplicaciones de Internet en dispositivos móviles

- Existen varios tipos de dispositivos móviles, pero los más populares son los teléfonos **móviles y las tablets**
  - Estos dispositivos suelen disponer de navegadores web completos, por lo que pueden acceder a cualquier aplicación web, pero debido a sus características (pequeña pantalla y control táctil) es recomendable que exista una versión adaptada de la aplicación web.
- “Diseño web adaptable” : Técnicas y herramientas que permiten que los elementos de la web se adapten al tamaño del dispositivo y su forma de interacción

# Aplicaciones de Internet en dispositivos móviles

- Las aplicaciones nativas de las plataformas móviles se implementan con **diferentes tecnologías**:
- **Android**: Java con librería móvil (No estándar)
- **iPhone y iPad**: ObjectiveC
- **Windows Phone**: C# o VisualBasic



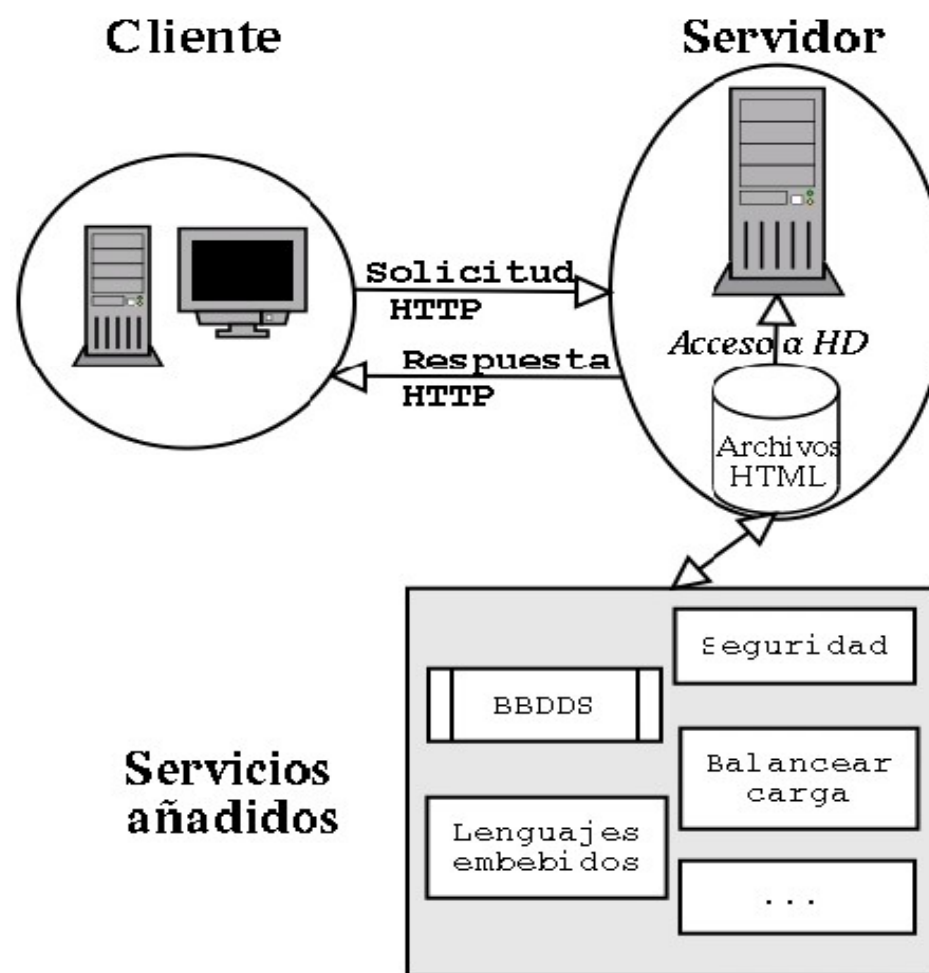
# Aplicaciones de Internet en dispositivos móviles

- Las tecnologías que se utilizan en el lado del cliente en el desarrollo web son **HTML, CSS y JavaScript**
- Se agrupan bajo el nombre de **HTML5**
- Permiten el diseño de **interfaces de usuario** atractivos, adaptados a diferentes resoluciones, interactivos, etc.
- Funcionan en una **gran cantidad de plataformas** gracias a los navegadores web (portables)

Estas tecnologías también se pueden usar para **implementar aplicaciones en los dispositivos móviles**. Estas aplicaciones se comportan como las **aplicaciones nativas**:

- Se publican en las tiendas de aplicaciones
- Tienen icono
- Pueden acceder a la cámara, acelerómetro, almacenamiento, etc...

# Arquitectura de un servidor de aplicaciones





# Servicios añadidos de los servidores de aplicaciones

- Generación de HTML
- Trabajo con bases de datos
- Funcionamiento multiproceso o multihilo
- Sesiones
- Lógica de negocio
- Seguridad
- Balanceo de carga

# OTRAS ARQUITECTURAS

- **Redes de pares (par-a-par o peer-to-peer => P2P)**
- **Arquitectura de tres-capas**, disponen de tres tipos de nodos:
  - Clientes que interactúan con los usuarios finales.
  - Servidores de aplicación que procesan los datos para los clientes.
  - Servidores de la base de datos que almacenan los datos para los servidores de aplicación

=> Mejora el balance la carga en los diversos servidores, es más escalable.

# Capa/Nivel

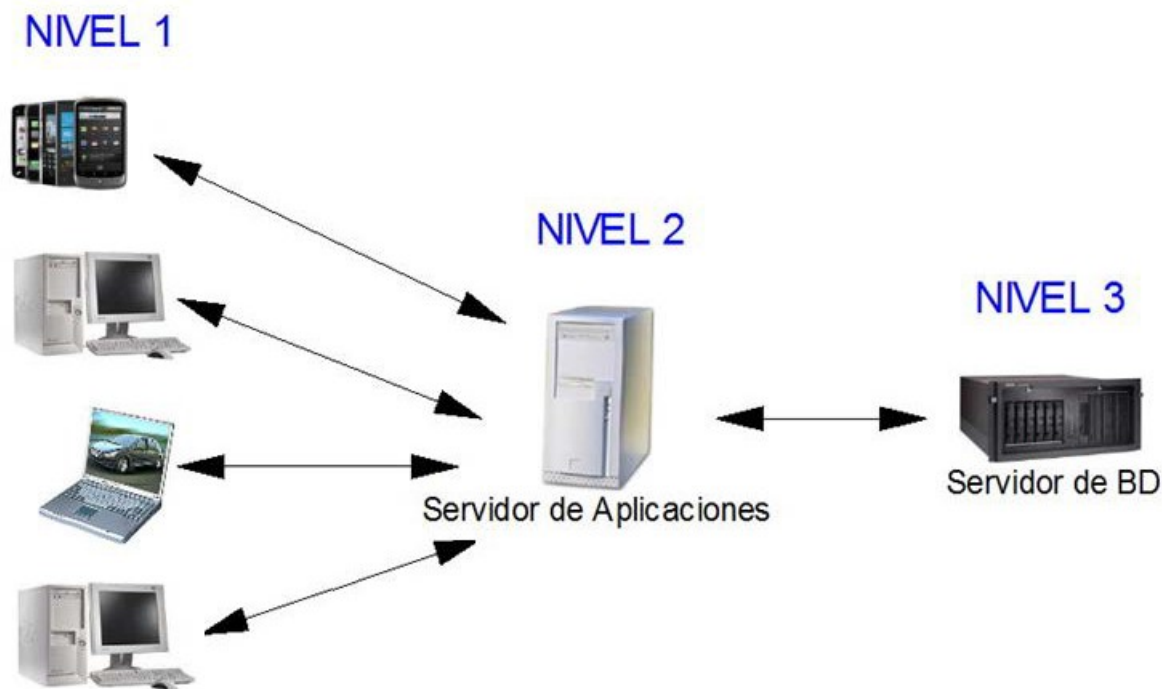
- El término "**capa**" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico:
  - Presentación.
  - Capa Lógica.
  - capa de Negocio (Datos).
- En cambio, el término "**nivel**" corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física.

# Modelo de tres capas

- **Capa de presentación:** es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario y le comunica la información.
- **Capa lógica:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso.
- **Capa de negocio (datos):** es donde residen los datos y es la encargada de acceder a los mismos.

# La arquitectura en tres niveles permite:

- ✓ Un mayor grado de flexibilidad.
- ✓ Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel.
- ✓ Mejor rendimiento, ya que las tareas se comparten entre servidores.
- ✓ Da lugar al diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).



# PROTOCOLOS DE APLICACIÓN MÁS UTILIZADOS

## EL PROTOCOLO HTTP

- El propósito del protocolo HTTP es permitir la transferencia de información (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL (Uniform Resource Locator, localizador uniforme de recursos).
- La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas: El navegador realiza una solicitud HTTP.
- El servidor procesa la solicitud y después envía una respuesta HTTP.

Hypertext Transfer Protocol (HTTP)	
Familia	Familia de protocolos de Internet
Función	Transferencia de hipertexto
Última versión	1.2
Puertos	80/TCP
Ubicación en la pila de protocolos	
Aplicación	HTTP
Transporte	TCP
Red	IP
Estándares	
RFC 1945 <a href="#">🔗</a> (HTTP/1.0, 1996)	
RFC 2616 <a href="#">🔗</a> (HTTP/1.1, 1999)	
RFC 2774 <a href="#">🔗</a> (HTTP/1.2, 2000)	

# EL PROTOCOLO HTTPS

- El protocolo seguro de Transferencia de hipertexto (HTTPS: Hiper Text Transfer Protocol Secure) es la versión segura del protocolo HTTP. La diferencia es que HTTPS permite realizar transacciones de forma segura. Por lo tanto, podremos desarrollar actividades de tipo *e-commerce*, *acceso a cuentas bancarias on line*, *tramites con la administración pública*, etc

Hypertext Transfer Protocol Secure (HTTPS)	
Familia	Familia de protocolos de Internet
Función	Transferencia segura de hipertexto
Puertos	443/TCP
Ubicación en la pila de protocolos	
Aplicación	HTTPS
Transporte	SSL/TLS
	TCP
Red	IP
Estándares	
RFC 2818 <a href="#">[editar datos en Wikidata]</a> – HTTP sobre TLS	

# EL PROTOCOLO FTP

- El protocolo FTP (File Transfer Protocol, Protocolo de transferencia de archivos) es, un protocolo para transferir archivos.
- El objetivo del protocolo FTP es: Permitir que equipos remotos puedan compartir archivos.
- Permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor.
- Permitir una transferencia de datos eficaz.

File Transfer Protocol (FTP)	
Familia	Familia de protocolos de Internet
Función	protocolo de transferencia de archivos
Puertos	20/TCP DATA Port 21/TCP Control Port
Ubicación en la pila de protocolos	
Aplicación	FTP
Transporte	TCP
Red	IP
Estándares	
FTP: RFC 959 <a href="#">(1985)</a>	
Extensiones de FTP para IPv6 y NATs: RFC 7178 <a href="#">(1998)</a>	



# EL PROTOCOLO SMTP

El protocolo SMTP (Simple Mail Transfer Protocol, Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

Simple Mail Transfer Protocol (SMTP)	
Familia	Familia de protocolos de Internet
Función	Envío de mensajes de correo-e
Puertos	25/TCP 587/TCP (alternativo para clientes de correo) 465/TCP (SMTPS)
Ubicación en la pila de protocolos	
Aplicación	SMTP
Transporte	TCP
Red	IP (IPv4 y IPv6)
Estándares	
RFC 821 <a href="#">🔗</a> (1982)	
RFC 2821 <a href="#">🔗</a> (2001)	
RFC 5321 <a href="#">🔗</a> (2008)	

# EL PROTOCOLO DNS

- **Domain Name System** o **DNS** (en español «Sistema de Nombres de Dominio») es un sistema de nomenclatura jerárquica para ordenadores, servicios o cualquier recurso conectado a Internet o a una red privada. Su función más importante, es traducir (resolver) nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red (Direcciones IP), esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

Domain Name System (DNS)	
Familia	Familia de protocolos de Internet
Función	Resolución de nombres de dominio
Puertos	53/UDP, 53/TCP
Ubicación en la pila de protocolos	
Aplicación	DNS
Transporte	TCP o UDP
Red	IP (IPv4, IPv6)
Estándares	
RFC 1034 <a href="#">🔗</a> (1987)	
RFC 1035 <a href="#">🔗</a> (1987)	

# Bibliografía

- <https://www.youtube.com/playlist?list=PLviLEHmst4ESCQmZ1ibNfgviOzBYuPYcG>
- <https://es.wikipedia.org/wiki/Internet>
- [http://es.wikipedia.org/wiki/Aplicaciones\\_Web](http://es.wikipedia.org/wiki/Aplicaciones_Web)
- <http://geeks.ms/blogs/sergiotarrillo/archive/2009/01/14/140214.aspx>
- [http://es.wikipedia.org/wiki/Aplicacion\\_cliente\\_servidor](http://es.wikipedia.org/wiki/Aplicacion_cliente_servidor)
- [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_por\\_capas](http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas)
- <https://www.youtube.com/watch?v=i4RE6dBAjH4>
- <https://www.youtube.com/watch?v=BWb6ri3ePew>
- <http://aprendiendoausarinternet.weebly.com/historia-del-internet.html>
- Despliegue de Aplicaciones Web. Ed. Garceta
- Implantación de Aplicaciones Web. Ed. Garceta