

Arquitectura Web

INTRODUCCION AL HTTP

Programa

- Un programa es una secuencia de instrucciones en algún lenguaje comprensible para la computadora con el fin de realizar una tarea específica.
- Es la implementación en un entorno específico en un lenguaje específico de una solución algorítmica a un problema

Aplicación

- Es un conjunto de programas.
- Interrelacionados con un fin específico
- Puede utilizar distintos lenguajes
- Es sinérgico
- Las aplicaciones están condicionadas por la infraestructura en la que se desarrollan y ejecutan
- Las aplicaciones pueden ser monolíticas, modulares, cliente/servidor, basadas en servicios, basadas en una arquitectura determinada, etc.

Aplicación Web

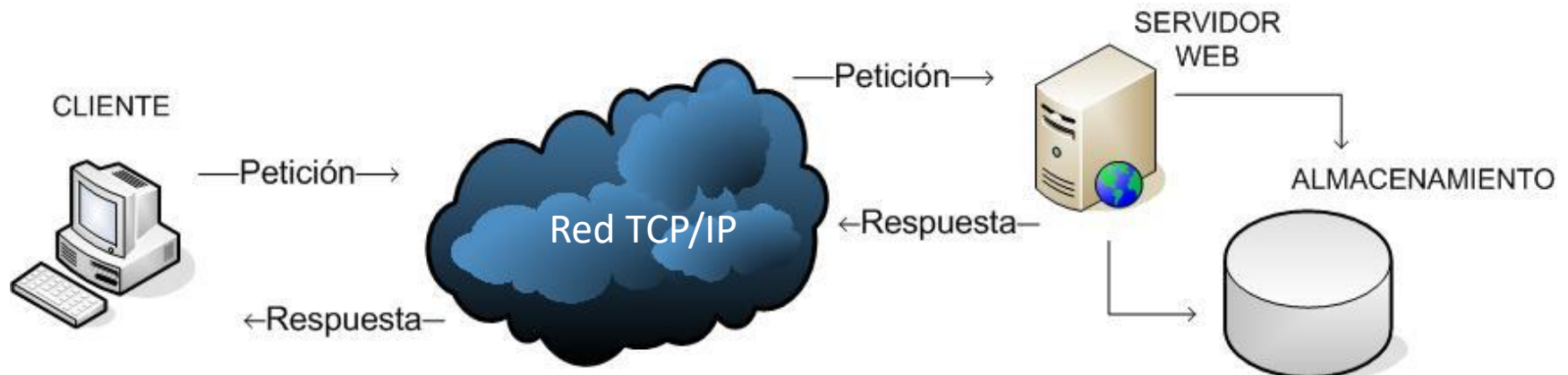
Es una construcción de software destinada a un uso general que utiliza una tecnología específica (WorldWideWeb) como medio ambiente de despliegue.

- Basa el intercambio de información entre la aplicación y el cliente en protocolos de Internet (Básicamente TCP/IP, HTTP y derivados)
- Codifica la GUI utilizando tecnología y protocolos disponibles en la WWW (HTML, CSS, JavaScript, etc.)
- Utiliza como mediador en la comunicación entre el Usuario y la Aplicación un navegador web o una aplicación específica basada en él.

Aplicación Web - Introducción

Web-based (web-enabled) application:

- Aplicación Cliente/Servidor: el cliente, el servidor y el protocolo de intercambio entre ellos ya están definidos (implementados) y son estándares.



Aplicación Web - Introducción

- Basada en cuatro elementos principales:

1.Red de Datos

- Red TCP/IP

2.Sistema único de identificación de recursos

- DNS/URL/URI

3.Ambientes aislados / Lenguajes Estándar

- Back-End / Front-End

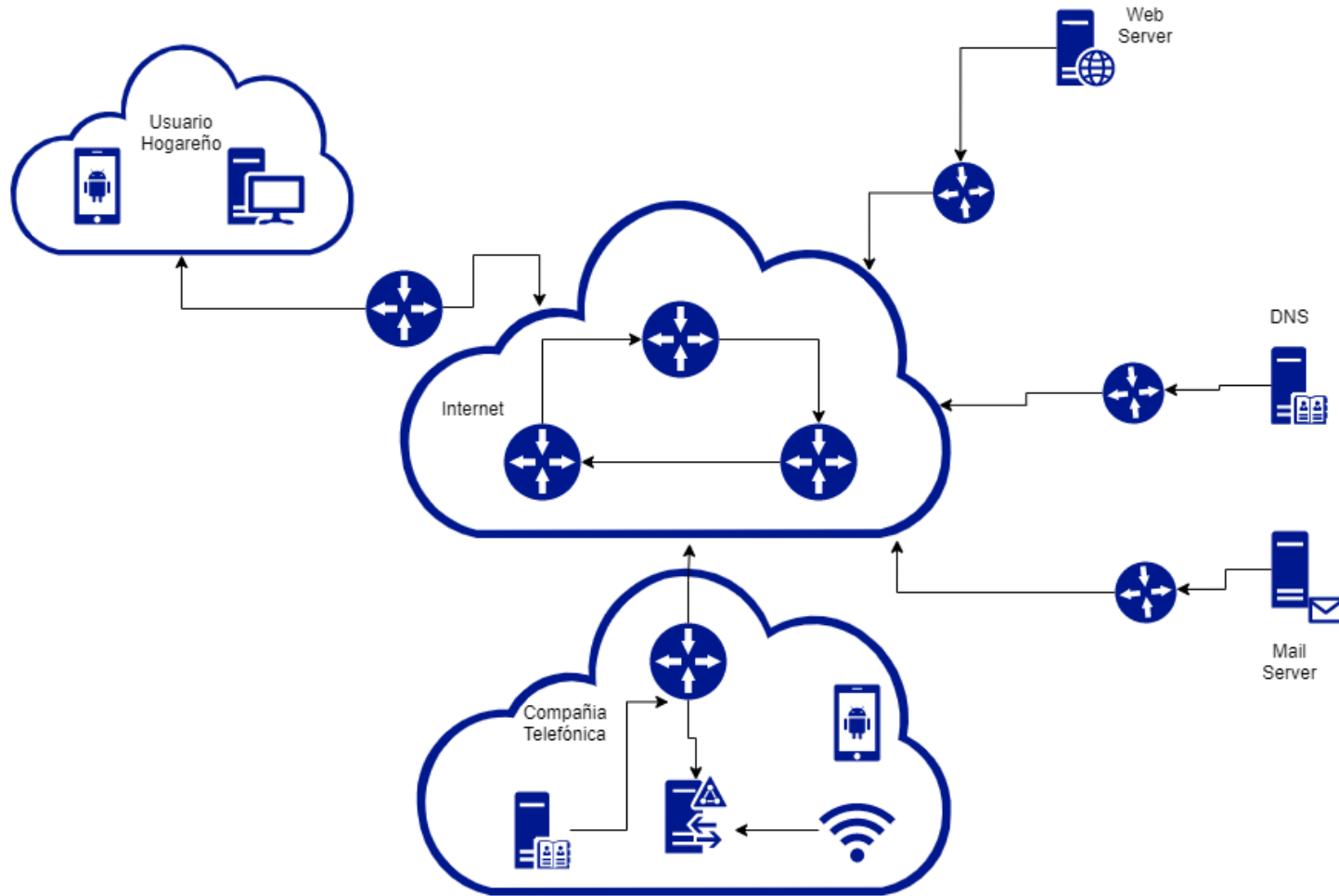
4.Servidores y Clientes HTTP

- Se comunica en HTTP, parsea HTML / CSS

Aplicación Web – 1.Red de datos

- Red de conmutación de paquetes basada en TCP/IP (Internet)
- Direccionamiento Único (Dirección IP)
- Gestión y encaminamiento en la red (Simplicidad y robustez)
- Protocolos estándar (*Stack* TCP/IP)
- Rutas
- Configuración basada en 3 parámetros:
 - Dirección IP
 - Puerta de enlace predeterminada
 - DNS

Aplicación Web – 1.Red de datos



Aplicación Web – 1.Red de datos

- Determinación de red local / externa

Par IP / Mascara: 192.168.0.1 / 255.255.255.0

IP Destino: 192.168.10.1

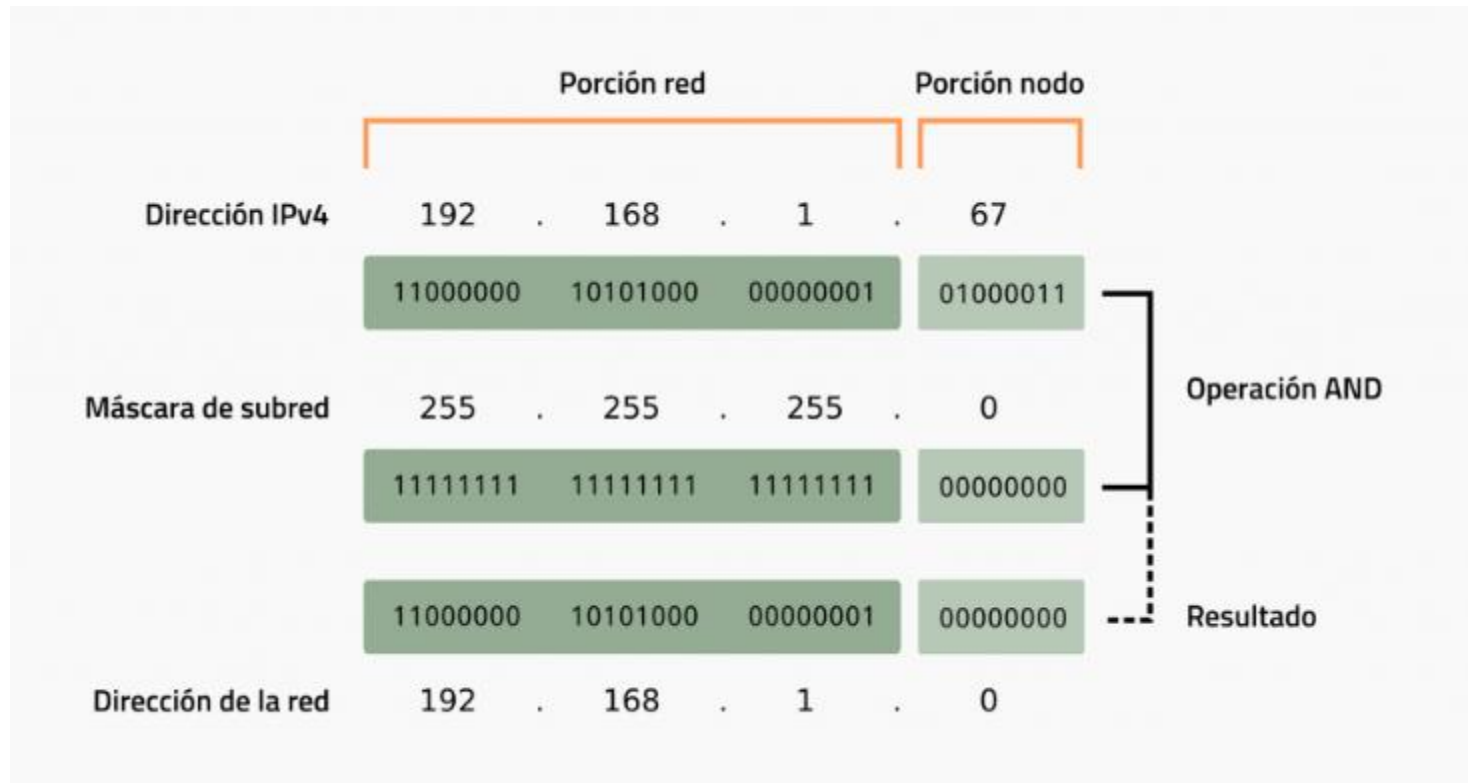
IP Destino binario: 11000000.10101000.00001010.00000001

IP Origen binario: 11000000.10101000.00000000.00000001

Mascara: 11111111.11111111.11111111.00000000

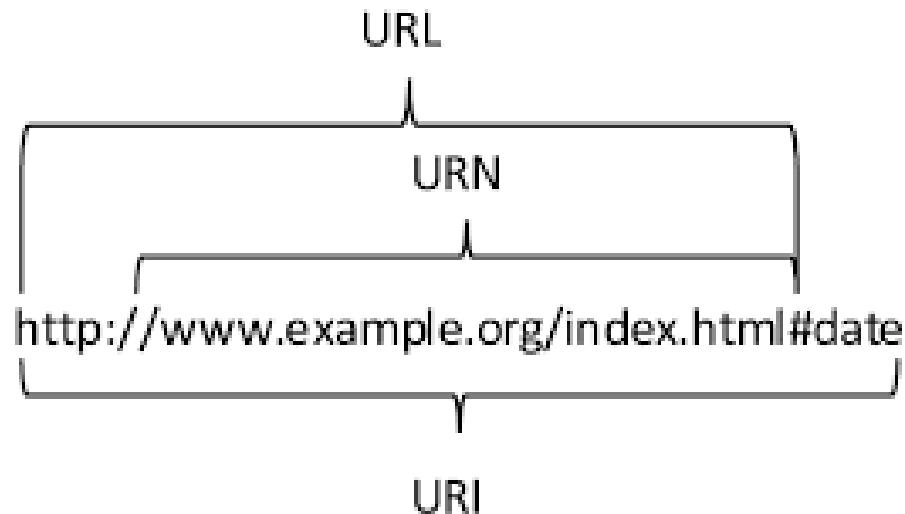
La IP no pertenece a la red alcanzable.

Aplicación Web – 1.Red de datos



Aplicación Web – 2.Sistema de Nombres

- Elementos principales:
 - DNS: Domain Name Service
 - URL: Uniform Resource Locator
 - URI: Uniform Resource Identifier
 - URN: Uniform Resource Name



Aplicación Web – 2.Sistema de Nombres

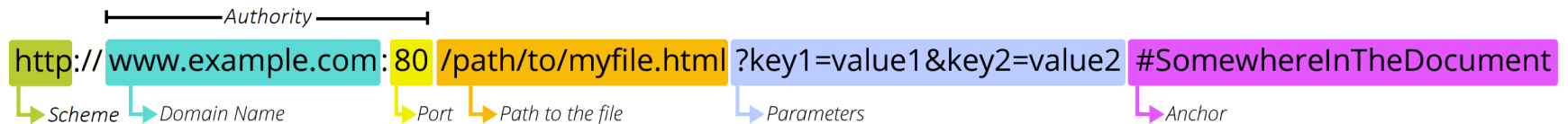
- URL: Uniform Resource Locator
 - Estándar de uso general
 - Es quien permite identificar un recurso único dentro de un servicio determinado
 - Definido en las RFC 1736, 1737 y 2396 entre otras
 - Una URL determinada en determinado período de tiempo es única para un cliente en todo el contexto de red donde se encuentre
 - Formato:

servicio://servidor:puerto/caminoDelRecurso/recurso

Ej: <https://www.unlam.edu.ar/index.php>

Aplicación Web – 2.Sistema de Nombres

- URL:




Aplicación Web – 2.Sistema de Nombres

- Esquema:

Ej: ftp, mailto, gopher, smb, etc.

http://www.example.com:80/path/




Scheme

Aplicación Web – 2.Sistema de Nombres

- Autoridad:

http://**www.example.com**:**80**/path/to/my



The diagram illustrates the components of a URL. The text 'http://www.example.com:80/path/to/my' is shown. The domain 'www.example.com' is highlighted in a light blue box, and the port '80' is highlighted in a yellow box. A light blue arrow points from the bottom of the blue box to the text 'Domain Name'. A yellow arrow points from the bottom of the yellow box to the text 'Port'.

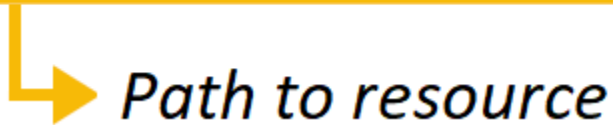
Domain Name

Port

Aplicación Web – 2.Sistema de Nombres

- Recurso:

m:80/path/to/myfile.html?key1=value1&



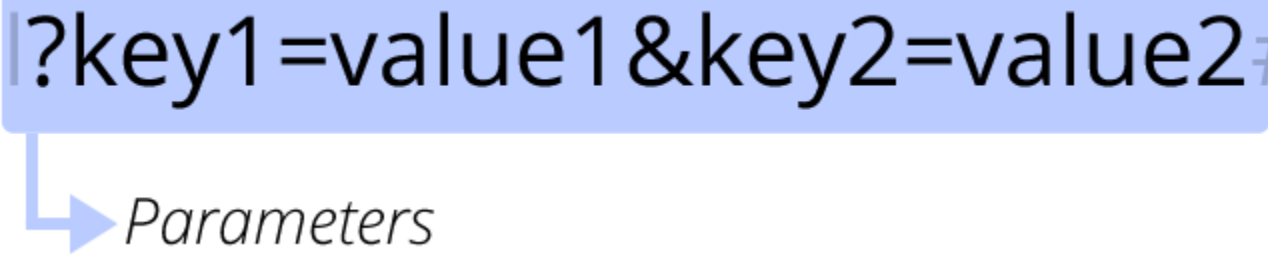
A yellow L-shaped arrow points from the text `/path/to/myfile.html` in the URL above to the text *Path to resource*.

Path to resource

Aplicación Web – 2.Sistema de Nombres

- Parámetros:

html?key1=value1&key2=value2#Some



The diagram illustrates the components of a URL. A light blue rectangular box highlights the portion of the URL starting with the question mark: `?key1=value1&key2=value2#Some`. A blue arrow originates from the bottom-left corner of this box, pointing downwards and then to the right towards the word *Parameters*, which is written in a grey, italicized font.

Aplicación Web – 2.Sistema de Nombres

- Anclas:

blue2#SomewhereInTheDocument



Aplicación Web – 2.Sistema de Nombres

- URL
 - Absolutas: Completa, Protocolo Implícito y Dominio Implícito
 - Relativas: Sub-recursos, Moverse por el árbol de recursos

Aplicación Web – 2.Sistema de Nombres

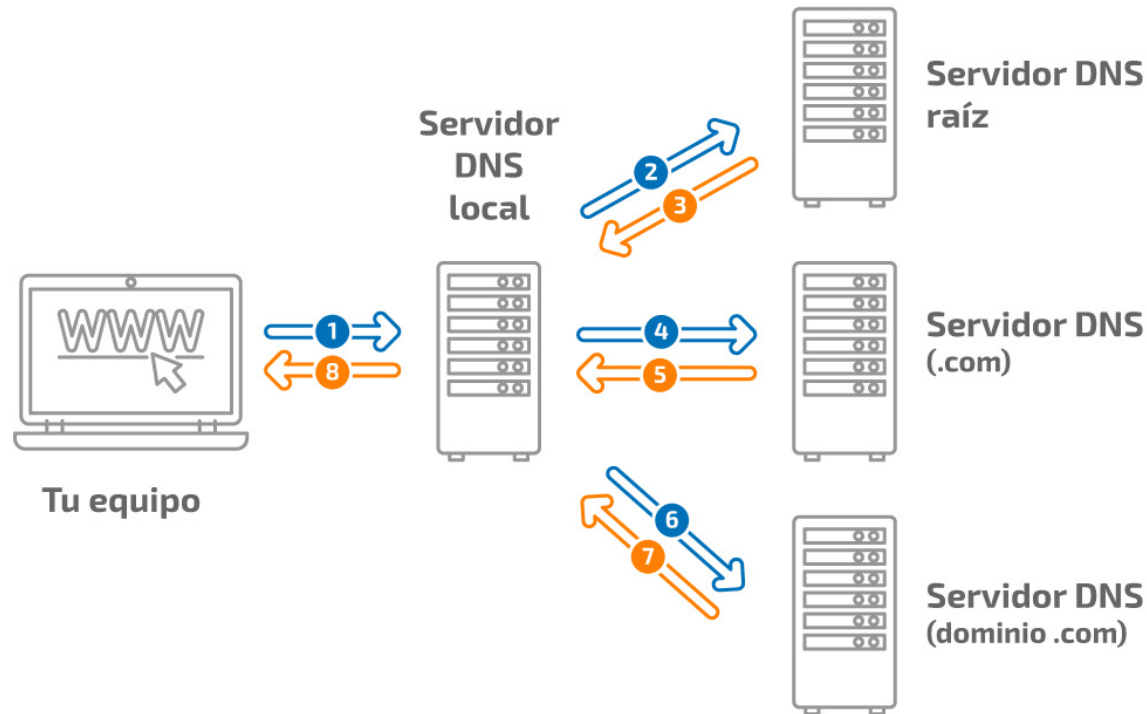
- DNS: Domain Name Service
 - Estándar de uso general
 - Definido en las RFC 881,1034 y 1035
 - Es quien convierte nombres de *host* en direcciones IP gestionables (encaminables) en la red
 - Sistema jerárquico y centralizado
 - Administrado por la ICANN (Corporación de Internet para la Asignación de Nombres y Números)

Aplicación Web – 2.Sistema de Nombres

- DNS: Domain Name Service
 - Define una serie de registros accesibles públicamente.
 - **A (Address), Registro de dirección.**
 - **AAAA (Address), Registro de dirección IPv6.**
 - **CAA (Certification Authority Authorization), Autorización de la Autoridad de Certificación.**
 - **CNAME (Canonical Name), Registro de nombre canónico.**
 - **MX (Mail eXchange), Registro de intercambio del correo.**
 - **PTR (Pointer), Registro de puntero.**
 - **SRV (Service record), Localizador de servicios.**
 - **TXT (Text), Registro de texto.**

Aplicación Web – 2.Sistema de Nombres

- DNS: Domain Name Service
 - Es un sistema jerárquico

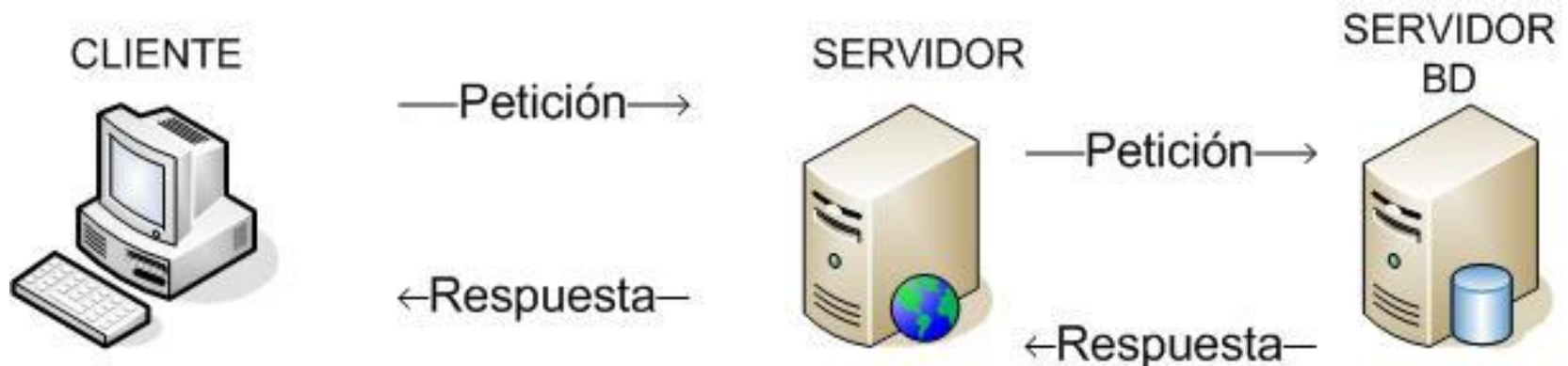


Aplicación Web – 2.Sistema de Nombres

Letra	Dirección IPv4	Dirección IPv6	Nombre antiguo	Operador	Software
<u>A</u>	198.41.0.4	2001:503:ba3e::2:30	ns.internic.net	<u>Verisign</u>	<u>BIND</u>
<u>B</u>	192.228.79.201	2001:478:65::53 (no en la zona raíz todavía)	ns1.isi.edu	<u>USC-ISI</u>	<u>BIND</u>
<u>C</u>	192.33.4.12	2001:500:2::c	c.psi.net	<u>Cogent Communications</u>	<u>BIND</u>
<u>D</u>	199.7.91.13 ⁹	2001:500:2d::d	terp.umd.edu	<u>Universidad de Maryland</u>	<u>BIND</u>
<u>E</u>	192.203.230.10	2001:500:a8::e	ns.nasa.gov	<u>NASA</u>	<u>BIND</u>
<u>F</u>	192.5.5.241	2001:500:2f::f	ns.isc.org	<u>Internet Systems Consortium</u>	<u>BIND 9¹⁰</u>
<u>G</u>	192.112.36.4	2001:500:12::d0d	ns.nic.ddn.mil	<u>Defense Information Systems Agency</u>	<u>BIND</u>
<u>H</u>	128.63.2.53	2001:500:1::803f:235	aos.arl.army.mil	<u>U.S. Army Research Lab</u>	<u>NSD</u>
<u>I</u>	192.36.148.17	2001:7fe::53	nic.nordu.net	<u>Netnod</u> (antes Autónoma)	<u>BIND</u>
<u>J</u>	192.58.128.30	2001:503:c27::2:30		<u>Verisign</u>	<u>BIND</u>
<u>K</u>	193.0.14.129	2001:7fd::1		<u>RIPE NCC</u>	<u>NSD</u>
<u>L</u>	199.7.83.42	2001:500:3::42		<u>ICANN</u>	<u>NSD</u>
<u>M</u>	202.12.27.33	2001:dc3::35		<u>Proyecto WIDE</u>	<u>BIND</u>

Aplicación Web – 3. Ambientes aislados

- La lógica de negocios y la persistencia de datos (Back-End) está físicamente separado del cliente que lo utiliza (Front-End) mediando una red TCP/IP entre ellos
- Back-End y Front-End no necesitan compartir ni tecnología ni arquitectura.
- Toda comunicación entre Back-End y Front-End se realiza mediante protocolo HTTP



Aplicación Web – 4. Servidor y Cliente HTTP

- HyperText Transfer Protocol (RFC 2616 HTTP 1.1)
- El propósito del protocolo HTTP es permitir la transferencia de archivos (recursos). entre un cliente (*Browser*) y un servidor (Servidor HTTP)
- Los recursos en el servidor se localizan mediante una cadena de caracteres estándar (URL)
- Es un protocolo de la Familia TCP/IP
- Se sitúa en el nivel 7 (aplicación) del OSI-RM (Open System Interconnection-Reference Model)
- Protocolo de transferencia de texto
- Cliente/Servidor
- Petición/Respuesta
- Connectionless & Stateless

Aplicación Web – 4.Servidor y Cliente HTTP

- 0.9 Solo transferencias (GET) a partir del 1.0, implementa cabeceras (POST)
- MIME – Types / Internet Content Types (codificación)
- Implementa un conjunto determinado de comandos:
 - GET (0.9)
 - POST (1.0)
 - HEAD (1.0)
 - PUT
 - DELETE
 - TRACE
 - OPTIONS
 - CONNECT
- Comandos mas utilizados: POST, GET y HEAD

Aplicación Web – 4.Servidor y Cliente HTTP

- GET HTTP/0.9 obtiene un recurso a partir de una URL
- HEAD HTTP/1.0 obtiene la cabecera descriptiva de recurso a partir de una URL
- POST HTTP/1.0 obtiene un recurso y puede enviar datos al servidor en el proceso
- PUT HTTP/1.1 pide al servidor guarde la información que se envía
- DELETE HTTP/1.1 para borrar un archivo en el servidor
- TRACE HTTP/1.1 para fines de depuración y seguimiento
- CONNECT HTTP/1.1 reservado para los *proxys* a fin de crear un tunel
- OPTIONS HTTP/1.1 lista las opciones de un determinado recurso

Aplicación Web – 4.Servidor y Cliente HTTP

- Petición HTTP
 - Encabezado + Línea en Blanco + Datos Adicionales
- Respuesta HTTP
 - Un código de estado sobre la petición + la información solicitada

Aplicación Web – 4.Servidor y Cliente HTTP

- GET
 - Clave GET + recurso + dos CRLF
 - Permite pasar parámetros dentro del URL
 - Limitado en tamaño (1024 *bytes*)
- HEAD
 - Sólo trae las cabeceras, no devuelve el cuerpo del documento
 - Se utiliza para sincronizar documentos, capacidades del servidor, etc.
- POST
 - Se pasan una serie de cabeceras seguidas de un cuerpo variable
 - Permite pasar parámetros en el cuerpo de la transferencia en vez de en la URL

Aplicación Web – 4.Servidor y Cliente HTTP

- La petición HTTP siempre devuelve un código del resultado de la petición:
 - 1xx indica un mensaje informativo solamente (100 continue)
 - 2xx indica que algo ha ocurrido (200 ok)
 - 3xx redirige al usuario a otra URL
 - 4xx indica un error en el cliente (404 host not found)
 - 5xx indica un error en el servidor

Aplicación Web – 4.Servidor y Cliente HTTP

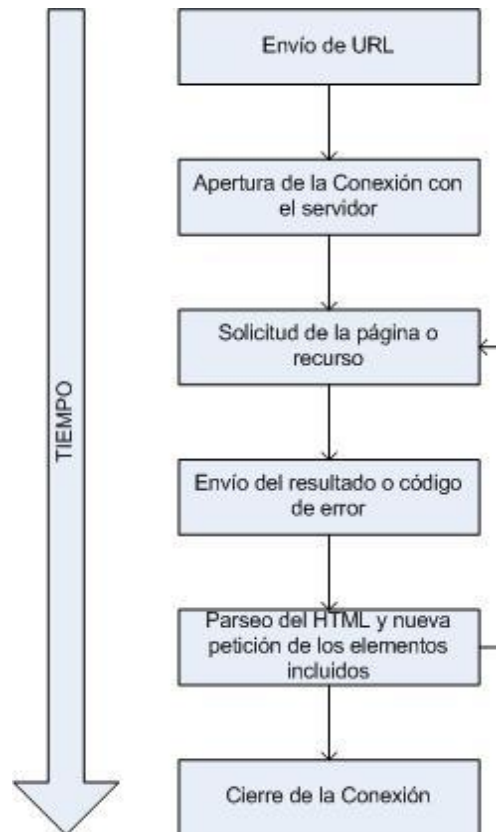
- El Cliente Web es quién gestiona las peticiones del usuario y la recepción de las páginas que provienen del servidor.
- Interpreta (*parsea y renderiza*) los documentos HTML y sus recursos. Las tecnologías más empleadas son:
 - HyperText Markup Language (HTML)
 - Cascading Style Sheets (CSS)
 - Lenguaje de script (JavaScript, VBScript, etc.)
- Es con quién interactúa el Usuario final

Aplicación Web – 4.Servidor y Cliente HTTP

- Servidor Web es la porción de software residente en un nodo determinado que espera peticiones: demonio (daemon) en Linux y servicio en servidores de Microsoft
- Implementa el protocolo HTTP y se encarga de atender las petición HTTP que convergen en ese nodo
- En la aplicación del servidor hay:
 - Páginas estáticas (documentos HTML, imágenes, media, etc.)
 - Pciones de código que al ser invocados se ejecutan en el entorno del servidor y dan como resultado un recurso que es devuelto al Cliente

Aplicación Web – 4.Servidor y Cliente HTTP

La secuencia de comunicación entre un Cliente y un servidor es la siguiente:



Aplicación Web – 4.Servidor y Cliente HTTP

La distribución de responsabilidades en una Aplicación Web debe ser:

