

# Programación Web 2

Introducción

# Académicos



Alfredo Paz Valderrama



Carlo Corrales Delgado

# ¿De qué trata el curso?

- Es la continuación de PWeb1, pero con **frameworks**

Los frameworks permiten el **rápido desarrollo** de aplicaciones evitando la construcción desde cero (**descubrir la rueda dos veces**).

Los frameworks también nos darán **mejores “costumbres”** sobre la manera de construir software, han sido hechos por expertos y aprenderemos de sus técnicas

- Donde se usó CSS, se usará **Bootstrap**
- Donde se usó Perl, se usará **Django**
- Donde se usó JavaScript, se usará **Angular**

# La nota final

- Eva. Continua 1 (23%)
- 1er Examen (10%)
  - 17-05-2020
- Eva. Continua 2 (23%)
- 2do Examen (10%)
  - 14-06-2020
- Eva. Continua 3 (24%)
- 3er Examen (10%)
  - 26-07-2020

# Reglas del Curso

1. No se aceptan tareas atrasadas,
2. En el curso se valora mucho la honestidad
3. En las tareas y trabajos se **calificará el proceso**, el producto final podría no funcionar y aún así tener buena nota
4. Si tiene dudas debe preguntar!
5. La prueba de entrada **no tendrá puntaje**, pero si no la hace a tiempo **no tendrá nota en sus tareas.**

# Actos que NO se consideran como deshonestos

- **Explicar lo que se pide en la tarea**, se puede pedir ayuda al profesor o los compañeros para entender lo que se pide en la tarea, pero siendo cuidadosos de no explicar la solución, sólo el enunciado de lo que se pide.
- **Explicar los temas o conceptos**, si algún tema o concepto no se entiende, fuera del horario de clase, se puede pedir ayuda al profesor o algún compañero, aunque siempre será preferible hacer la pregunta en clase para que todos entiendan mejor el tema.
- **Usar apuntes en las evaluaciones o exámenes**, estos apuntes podrán ayudar a recordar comandos, códigos, etc.
- **Estudiar en grupo**, revisando los temas tratados en clase.

# Actos que se consideran deshonestos

- **Copiar la solución de otro**, esto incluye mirar el código de un compañero o buscar soluciones o código en Internet
- **Compartir código fuente**, copiar, cambiar de nombre a las variables, mostrar el código a un compañero, descargar el código de Internet, explicar el código a un compañero. Tener cuidado de no dejar copias de las tareas en lugares públicos.
- **Consultoría**, recibir ayuda en la solución de la tarea, esta puede ser en persona, por un compañero de años superiores, por foros de discusión en Internet, etc.

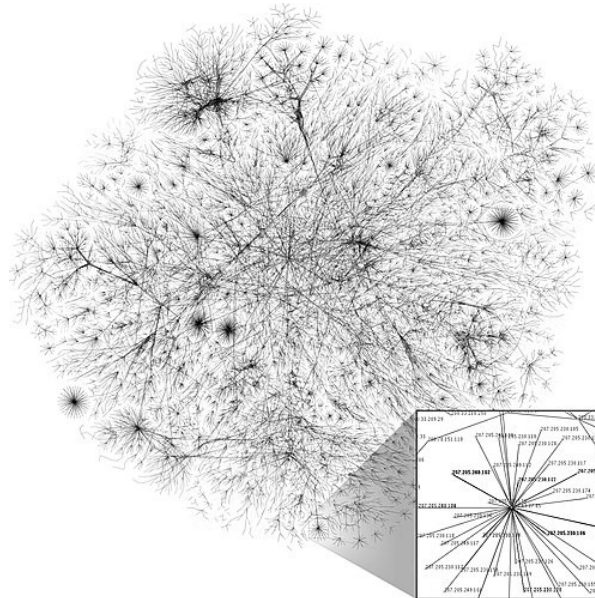
# Actos que se consideran deshonestos

- **Realizar los trabajos individuales en grupo**, las tareas pueden tener soluciones diversas, si estas son individuales no deben reunirse para hacerlas.
- **Realizar las tareas grupales de manera individual**, qué sólo un compañero haga toda la tarea del grupo, que cada integrante del grupo haga una parte de la tarea, pero que no tenga idea de las demás partes. Las tareas en grupo deben ser hechas en grupo, por lo que se requiere coordinación, no sólo en la distribución del trabajo, sino en la solución de los problemas que se puedan presentar. El grupo debe trabajar como un equipo.



# Internet

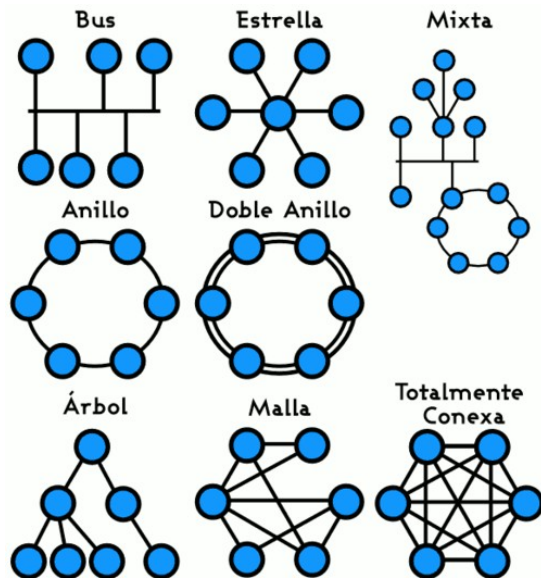
Una red de redes o varias internet juntas (note la minúscula en la palabra)



Fuente: wikipedia

# Topologías

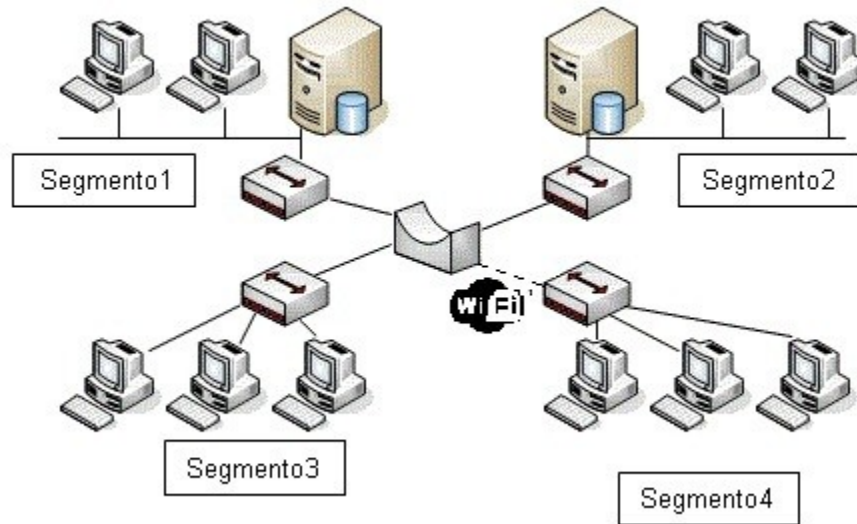
La topología de red se define como el **mapa físico o lógico de una red** para intercambiar datos. Se transmiten **paquetes de red** como señales eléctricas!



Fuente: wikipedia

# Bridge

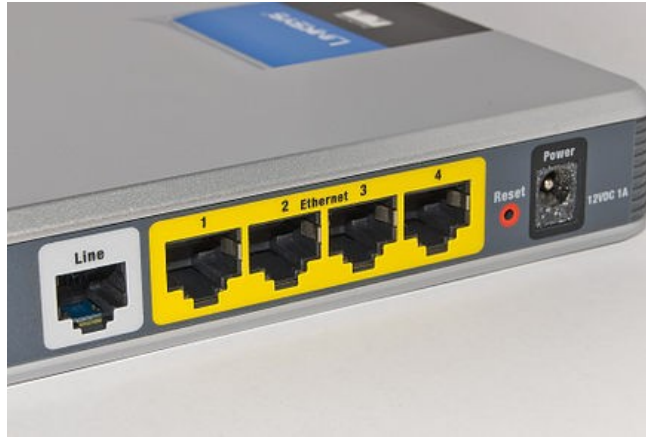
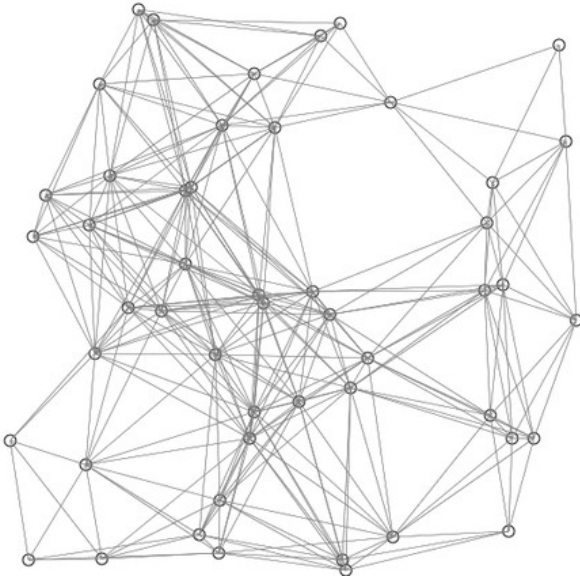
Un bridge conecta segmentos de red formando una sola subred (permite conexión entre equipos sin necesidad de routers).



Fuente: wikipedia

# Router

Envía los paquetes de red por el camino o ruta más adecuada en cada momento



Fuente: wikipedia

# Modem

Un módem (del inglés modem, acrónimo de modulator demodulator; pl. módems) es un dispositivo que convierte las señales digitales en analógicas (modulación) y viceversa (desmodulación)



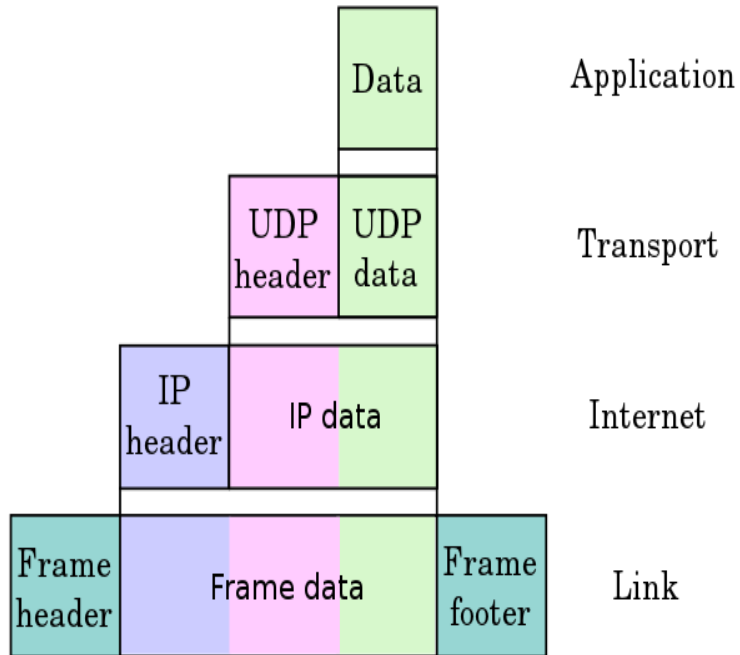
# Protocolos de comunicaciones

Reglas que permiten que dos o más entidades de un sistema se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física.

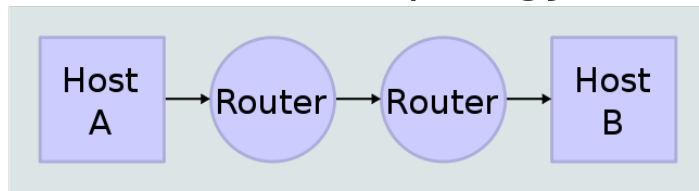
Por ejemplo el protocolo para saludar a un amigo o empezar una conversación con alguien.



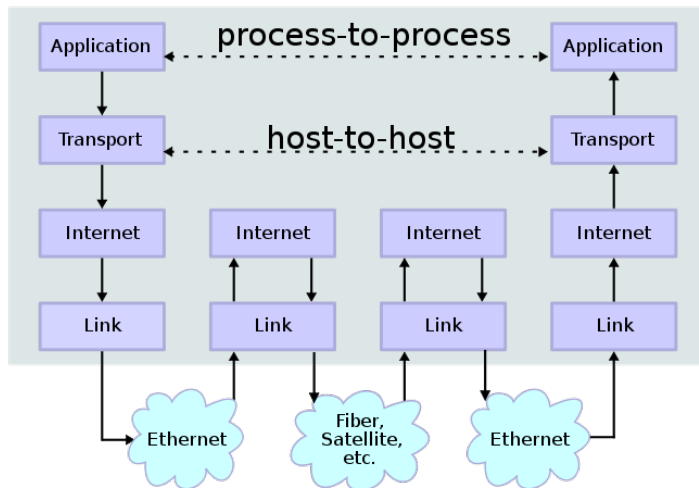
# Capas de abstracción



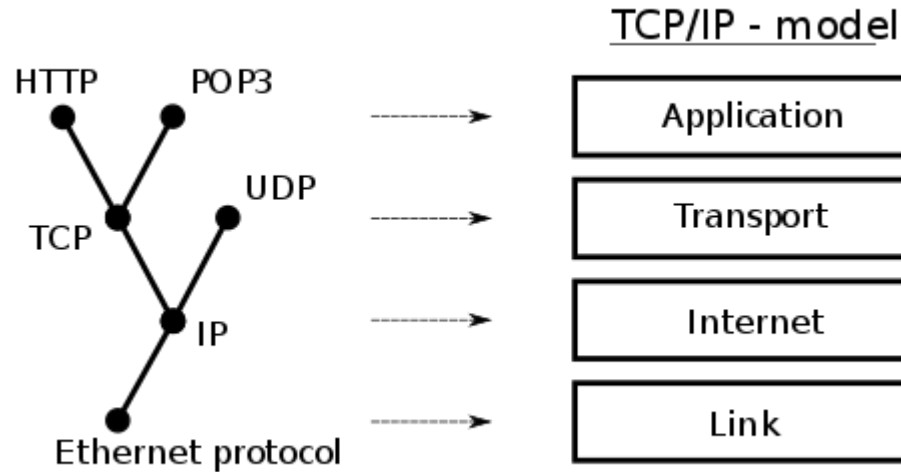
## Network Topology



## Data Flow



# Ejemplos





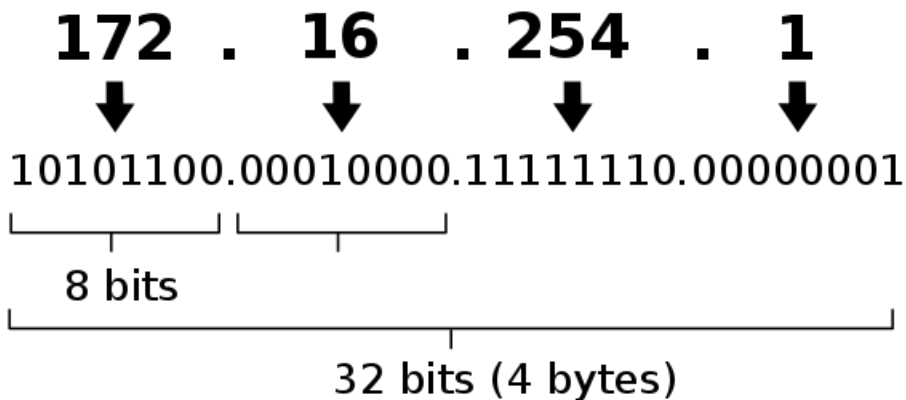
# IPv4

- Desarrollado por ARPANET en 1983
- Su sucesor IPv6, desarrollado en el año 1998 por la Internet Engineering Task Force (IETF) ratificado en el 2017
- Utiliza un sistema de **direccionamiento lógico** y realiza el **enrutamiento**, que es el reenvío de paquetes desde un host de origen al siguiente enrutador que está un salto más cerca del host de destino previsto en otra red.
- No garantiza la entrega, ni integridad de los datos
- Ejemplo de implementación en el lenguaje C  
[http://lxr.linux.no/linux+v2.6.37/net/ipv4/tcp\\_ipv4.c](http://lxr.linux.no/linux+v2.6.37/net/ipv4/tcp_ipv4.c)

# IPv4 - Direccionamiento

Las direcciones son de 32 bits, cada aparato conectado debe tener un número que lo identifica de manera única

IPv4 address in dotted-decimal notation



# IPv4 - Mascara de Red

	Binary form	Dot-decimal notation
IP address	11000000.00000000.00000010.10000010	192.0.2.130
Subnet mask	11111111.11111111.11111111.00000000	255.255.255.0
Network prefix	11000000.00000000.00000010.00000000	192.0.2.0
Host identifier	00000000.00000000.00000000.10000010	0.0.0.130

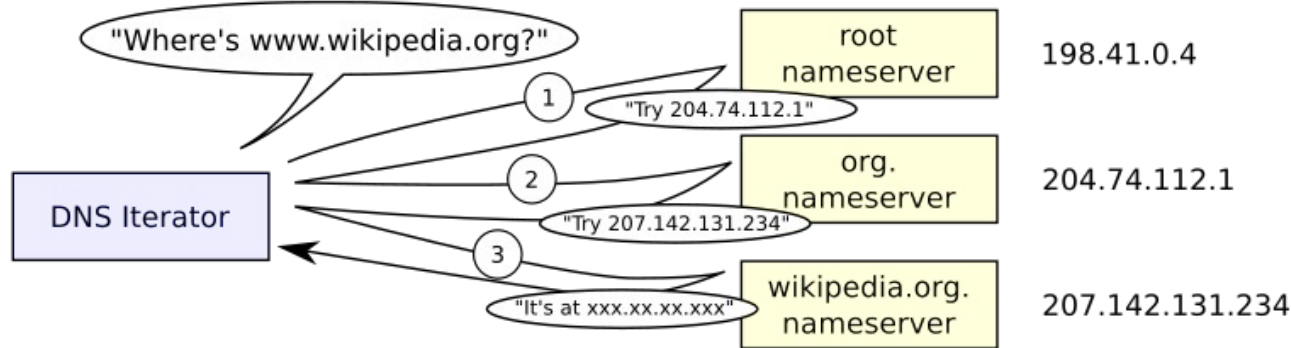
# IPv4 - Direcciones privadas

Name	CIDR block	Address range	Number of addresses
24-bit block	10.0.0.0/8	10.0.0.0 – 10.255.255.255	16 777 216
20-bit block	172.16.0.0/12	172.16.0.0 – 172.31.255.255	1 048 576
16-bit block	192.168.0.0/16	192.168.0.0 – 192.168.255.255	65 536

# Domain Name System

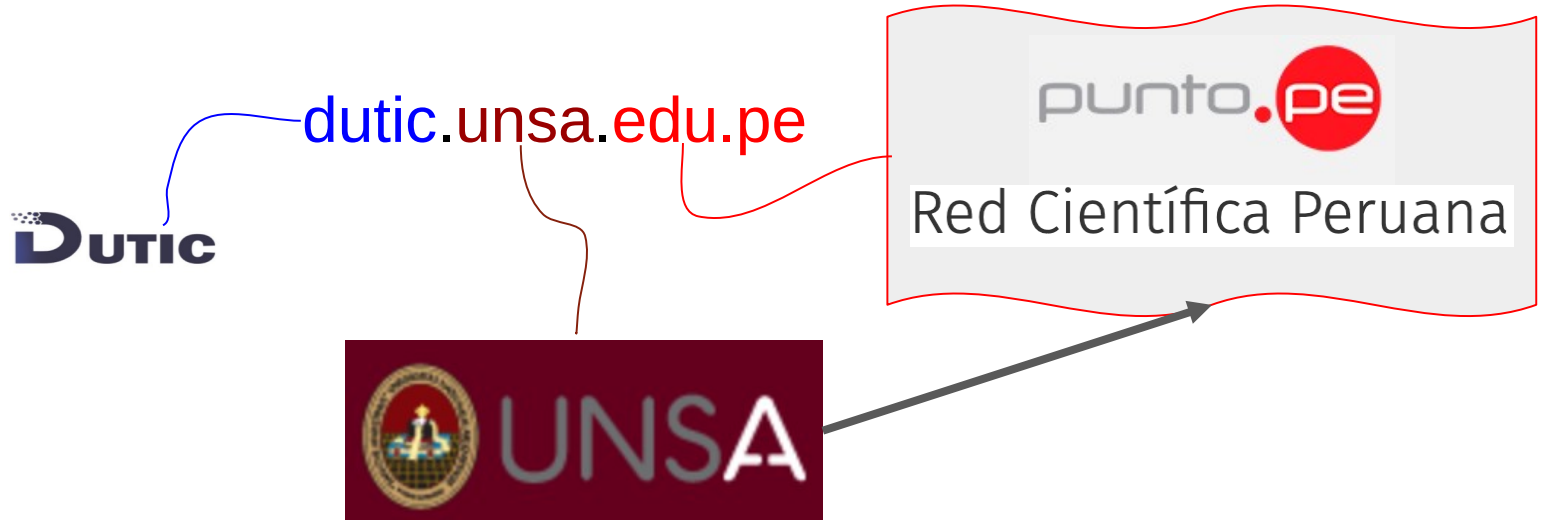
Traduce nombres de dominio memorizados a direcciones IP numéricas necesarias para localizar e identificar servicios y dispositivos

Es un sistema jerárquico y descentralizado



# Domain Name

Un nombre de dominio es una cadena de identificación que define un ámbito de autonomía administrativa, autoridad o control dentro de Internet.

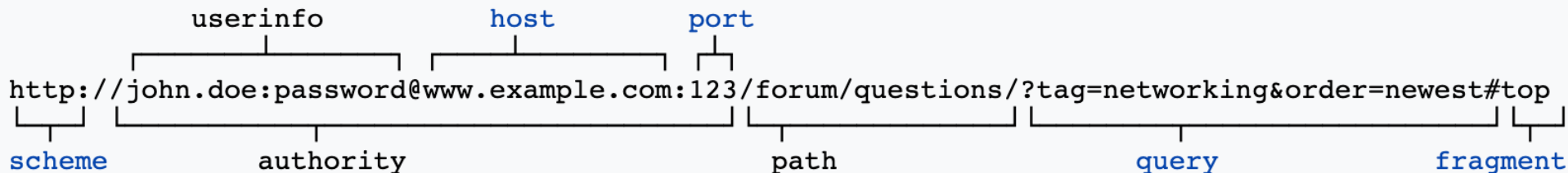


# Hypertext Transfer Protocol

- El Protocolo de transferencia de hipertexto (HTTP) es un **protocolo de aplicación** para sistemas de información distribuidos, colaborativos e hipermedia.
- El desarrollo de HTTP fue iniciado por **Tim Berners-Lee** en el CERN en 1989
- HTTP funciona como un protocolo de **solicitud-respuesta** en el modelo cliente-servidor, **no tiene estado**
- Un navegador web es un ejemplo de un agente de usuario (UA).
- Otros tipos de **agente de usuario** crawlers, navegadores de voz, aplicaciones móviles.

# Uniform Resource Locators (URLs)

Los recursos HTTP se identifican y ubican en la red mediante localizadores uniformes de recursos (URL), utilizando los esquemas de identificadores uniformes de recursos (URI) http y https.





# Métodos de petición

El protocolo define algunos métodos que podrían indicar que se desea alguna determinada acción, sin embargo esto podría variar según la implementación que se tenga en el servidor.

GET, HEAD, OPTIONS y TRACE, se definen, por convención, como seguros, lo que significa que están destinados solo a la recuperación de información y no deben cambiar el estado del servidor.

POST, PUT, DELETE y PATCH están destinados a acciones que pueden causar **efectos secundarios en el servidor** (cambio de estado) o **efectos secundarios externos** como transacciones financieras o transmisión de correo electrónico.

```
josh@blackbox:~$ telnet en.wikipedia.org 80
Trying 208.80.152.2...
Connected to rr.pmtpa.wikimedia.org.
Escape character is '^]'.
GET /wiki/Main_Page http/1.1
Host: en.wikipedia.org
```

Request

```
HTTP/1.0 200 OK
Date: Thu, 03 Jul 2008 11:12:06 GMT
Server: Apache
X-Powered-By: PHP/5.2.5
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Content-Language: en
Vary: Accept-Encoding, Cookie
X-Vary-Options: Accept-Encoding;list-contains=gzip, Cookie;string-contains=enwikiToken;string-contains=enwikiLoggedOut;string-contains=enwiki_session;
string-contains=centralauth_Token;string-contains=centralauth_Session;string-contains=centralauth_LoggedOut
Last-Modified: Thu, 03 Jul 2008 10:44:34 GMT
Content-Length: 54218
Content-Type: text/html; charset=utf-8
X-Cache: HIT from sq39.wikimedia.org
X-Cache-Lookup: HIT from sq39.wikimedia.org:3128
Age: 3
X-Cache: HIT from sq38.wikimedia.org
X-Cache-Lookup: HIT from sq38.wikimedia.org:80
Via: 1.0 sq39.wikimedia.org:3128 (squid/2.6.STABLE18), 1.0 sq38.wikimedia.org:80 (squid/2.6.STABLE18)
Connection: close
```

Response headers

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="keywords" content="Main Page,1778,1844,1863,1938,1980 Summer Olympics,2008,2008 Guizhou riot,2008 Jerusal
    ...
    ... This content has been removed to save space
    ...
    "Non-profit organization">nonprofit</a> <a href="http://en.wikipedia.org/wiki/Charitable_organization" title="Charitable organization">charity</a>.<br />
    </li>
    <li id="privacy"><a href="http://wikimediafoundation.org/wiki/Privacy_policy" title="wikimedia:Privacy policy">Privacy policy</a></li>
    <li id="about"><a href="/wiki/Wikipedia:About" title="Wikipedia:About">About Wikipedia</a></li>
    <li id="disclaimer"><a href="/wiki/Wikipedia:General_disclaimer" title="Wikipedia:General disclaimer">Disclaimers</a>
  </li>
</ul>
</div>
</div>
<script type="text/javascript">if (window.runOnloadHook) runOnloadHook();</script>
<!-- Served by srv93 in 0.050 secs. --></body></html>
Connection closed by foreign host.
josh@blackbox:~$
```

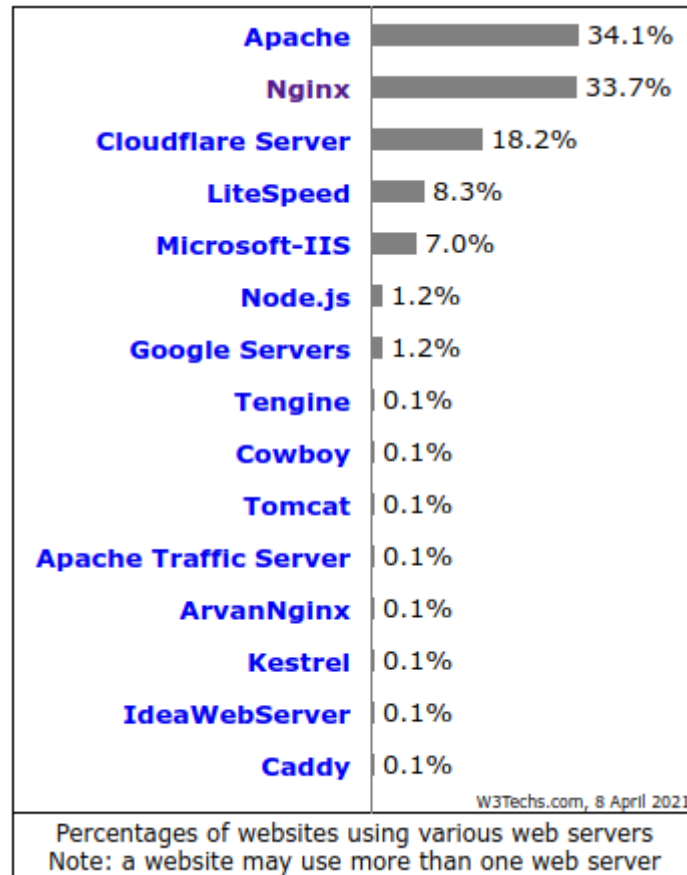
Response body

# Servidores Web

Un servidor es un programa de computadora o un dispositivo que proporciona funcionalidad para otros programas o dispositivos, llamados "clientes".

Los clientes Web pueden ser Browsers como Chrome, Firefox, etc.

[https://w3techs.com/technologies/overview/web\\_server](https://w3techs.com/technologies/overview/web_server)

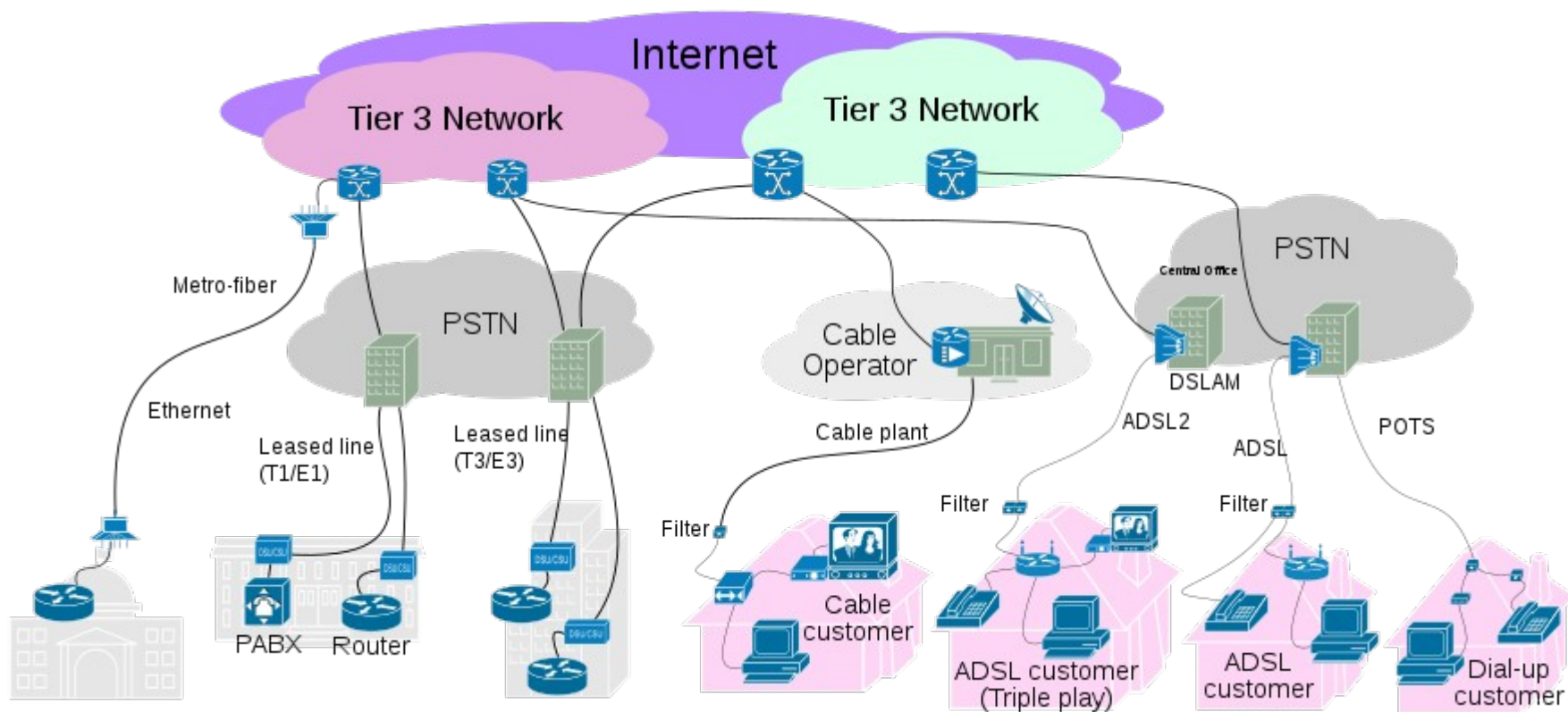


# Puertos

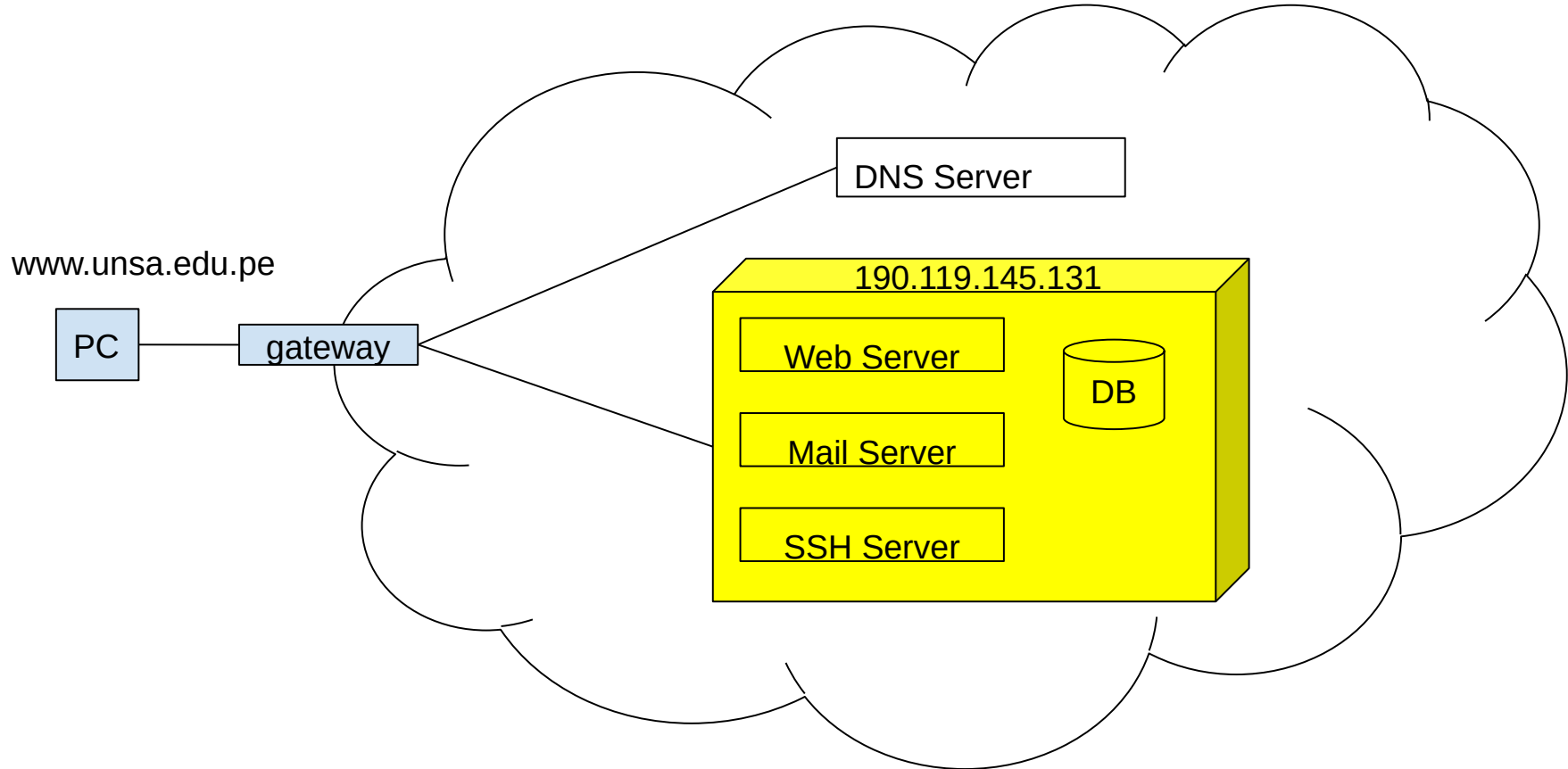
En una computadora, pueden haber varios servidores funcionando, cada uno escuchará las peticiones en un “puerto” distinto y entenderá un protocolo específico. el sistema operativo se encargará de controlar el acceso.

Port number	Assignment
20	<a href="#">File Transfer Protocol</a> (FTP) Data Transfer
21	<a href="#">File Transfer Protocol</a> (FTP) Command Control
22	<a href="#">Secure Shell</a> (SSH) Secure Login
23	<a href="#">Telnet</a> remote login service, unencrypted text messages
25	<a href="#">Simple Mail Transfer Protocol</a> (SMTP) E-mail routing
53	<a href="#">Domain Name System</a> (DNS) service
67, 68	<a href="#">Dynamic Host Configuration Protocol</a> (DHCP)
80	<a href="#">Hypertext Transfer Protocol</a> (HTTP) used in the <a href="#">World Wide Web</a>

# Internet Server Provider



# Comunicación entre cliente y servidor



# Common Gateway Interface

Common Gateway Interface (CGI) es una especificación de interfaz para servidores web para ejecutar programas como aplicaciones de consola (línea de comandos) que se ejecutan en un servidor que genera páginas web dinámicamente.

Un CGI combinado con un formulario y una base de datos permite resolver consultas de manera simple

Los CGIs no son escalables, por lo que actualmente se han desarrollado varias alternativas como los contenedores de Java.

# CGI en Perl

```
#!/usr/bin/perl
```

```
=head1 DESCRIPTION
```

```
printenv – a CGI program that just prints its environment
```

```
=cut
```

```
print "Content-Type: text/plain\n\n";
```

```
for my $var ( sort keys %ENV ) {  
    printf "%s = \"%s\"\n", $var, $ENV{$var};  
}
```



# CGI en Python

```
#!/usr/bin/python3
```

```
import cgi, cgitb
```

```
cgitb.enable()
```

```
input_data = cgi.FieldStorage()
```

```
print('Content-Type: text/html') # HTML is following
```

```
print('') # Leave a blank line
```

```
print('<h1>Addition Results</h1>')
```

```
try:
```

```
    num1 = int(input_data["num1"].value)
```

```
    num2 = int(input_data["num2"].value)
```

```
except:
```

```
    print('<output>Sorry, the script cannot turn your inputs into numbers (integers).</output>')
```

```
    raise SystemExit(1)
```

```
print('<output>{0} + {1} = {2}</output>'.format(num1, num2, num1 + num2))
```

# Hypertext Markup Language (HTML)

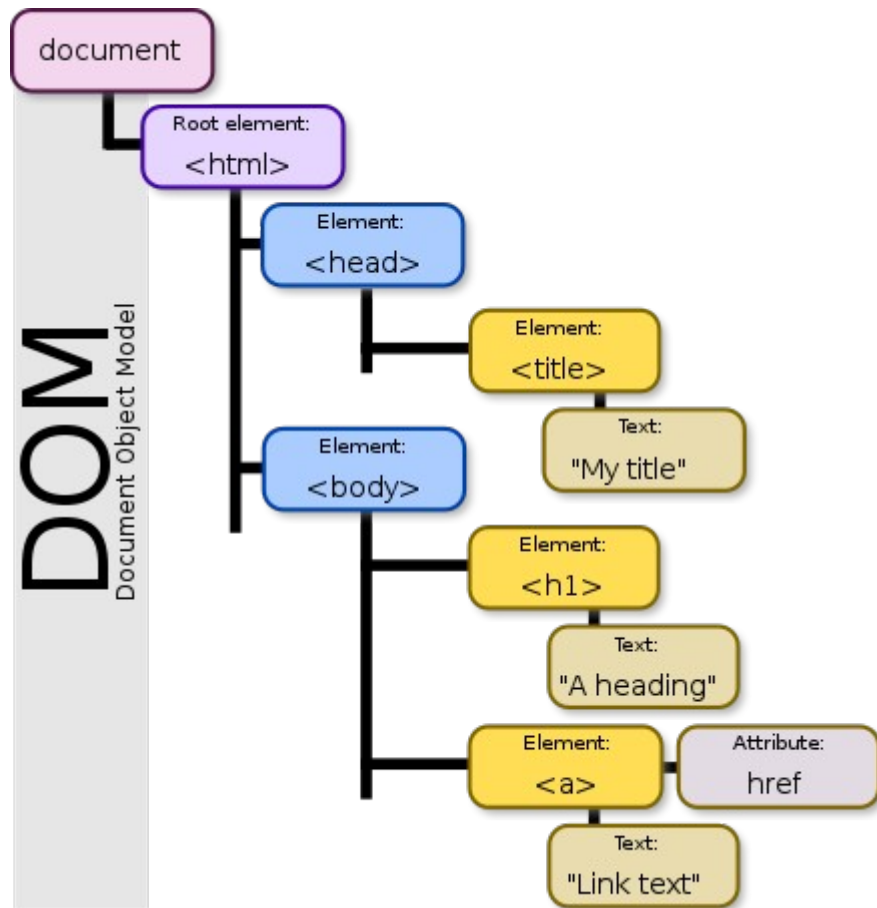
El lenguaje de marcado de hipertexto (HTML) es el lenguaje de marcado estándar para documentos diseñados para mostrarse en un navegador web. Puede ser asistido por tecnologías como Cascading Style Sheets (CSS) y lenguajes de script como JavaScript.

La especificación HTML5 se adoptó como el punto de partida del trabajo del nuevo grupo de trabajo HTML del World Wide Web Consortium (W3C) en 2007.

Las API y el Modelo de objetos de documento (DOM) ahora son partes fundamentales de la especificación HTML5

# Document Object Model

El Document Object Model (DOM) es una interfaz multiplataforma e independiente del lenguaje que trata un documento HTML como una estructura de árbol en la que cada nodo es un objeto que representa una parte del documento.



# Cascading Style Sheets

Cascading Style Sheets (CSS) es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en un lenguaje de marcado como HTML

Separar el estilo (CSS) del contenido (HTML), tiene la ventaja de que se pueda cambiar la apariencia sin hacer mayores o quizá ningún cambio en la lógica de los programas.

Los frameworks CSS son bibliotecas preparadas para permitir un estilo de páginas web más fácil y más compatible con los estándares utilizando el lenguaje de hojas de estilo en cascada: Foundation, Blueprint, Bootstrap, Cascade Framework y Materialise.

# Preguntas