

Módulo 2 - Tema 0

Introducción al Desarrollo Web



**CODE
SPACE**
ACADEMY

Índice

- ¿Qué es un programa?
- ¿Qué es un lenguaje de programación?
- Tipos de lenguajes
- Algoritmos
- Diagramas de flujo
- ¿Qué es internet? – ISP
- URLs, dominios, IPs y resolución DNS
- Modelo Cliente - Servidor
- Protocolo HTTP
- Front-End vs Back-End

1. ¿Qué es un programa? - Definición

Un programa informático es una **secuencia de instrucciones escritas** para realizar una **tarea específica** en una computadora.

Este ordenador requiere programas para funcionar, y por lo general, ejecuta las instrucciones del programa en un procesador central (CPU o *Central Processing Unit*).

El programa tiene un **formato ejecutable** que la computadora puede utilizar directamente para ejecutar las instrucciones y a su vez, también dispone de su **código fuente**, que es legible y nos permite como programadores estudiar y desarrollar todas sus instrucciones.

Una colección de programas de computadora y datos relacionados se conoce como **software**: código fuente, binario o ejecutable, la documentación del código, los datos a procesar e incluso la información de usuario

forman parte del software: es decir, abarca todo lo intangible, todo lo «no físico» relacionado.

```
.APK → Android Package File
.JAR → Java Archive
.RUN → Linux Executable File
.XBE → Xbox Executable File
.CMD → Windows Command File
.IPA → iOS Application
.EXE → Windows Executable File
.BIN → Unix Executable File
.BIN → Generic Binary Executable File
.ELF → Nintendo Wii Game File
.BAT → DOS Batch File
.APP → macOS Application
.COM → DOS Command File
```

2. ¿Qué es un lenguaje de programación?



Es un conjunto de reglas y sintaxis con la que podemos escribir un programa determinado. Hay muchos tipos de lenguaje de programación. Se podría hacer un símil entre un lenguaje de programación y un idioma.

3. Tipos de lenguajes

Los lenguajes de programación pueden clasificarse por **nivel de abstracción**:

Bajo nivel:

- Más próximos a la arquitectura del hardware. **Dependiente** de la máquina, por lo que no pueden usarse en otras máquinas.
- Estos lenguajes son mucho más **rápidos** que los lenguajes de alto nivel.
- Muy **difíciles de manejar** y usar, además de tener códigos fuente enormes.



3. Tipos de lenguajes

```
// Life motto  
if (sad) {  
    sad = false;  
    beAwesome();  
}
```

Alto nivel:

- Más cercanos a los programadores y usuarios, más **cercanos al lenguaje natural**.
- Este lenguaje es independiente de la arquitectura del ordenador.
- Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de la máquina/s para la que están diseñando el programa. Tan solo necesitan un traductor que entiendan tanto el código fuente como las características de la máquina.

3. Tipos de lenguajes

También se pueden clasificar según el **tipo** de traductor (para lenguajes de alto nivel):

2.1 Compilados:

- Analiza el programa fuente y lo **traduce** a uno equivalente escrito en otro lenguaje (por ejemplo, en el lenguaje de la máquina).
- Programas **más rápidos y eficientes**, ya que el análisis del lenguaje fuente se hace una sola vez, durante la generación del programa equivalente.

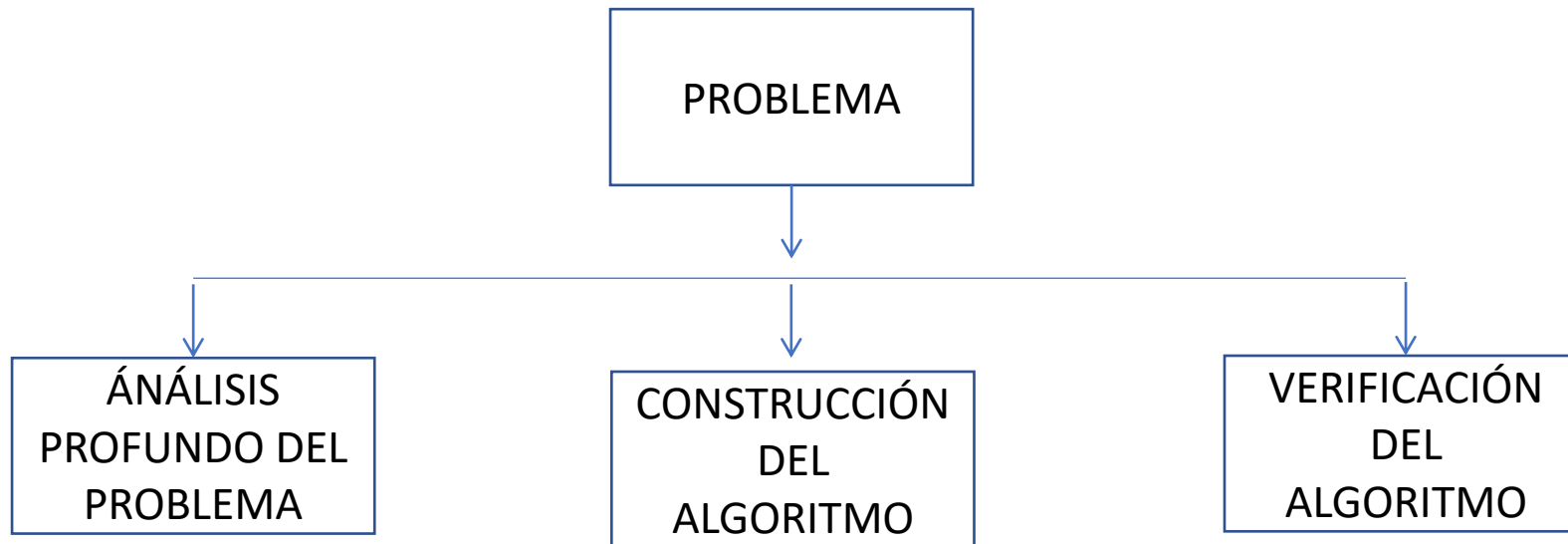
2.2 Interpretados:

- Analiza el programa fuente y **lo ejecuta directamente**, sin generar ningún código equivalente.
- La ejecución de un programa puede interrumpirse en cualquier momento para estudiar el entorno (depuración). Además, el programa puede modificarse sobre la marcha.

4. Algoritmos

Definición: Conjunto de pasos, procedimientos o acciones que nos permiten alcanzar un resultado o resolver un problema. Ejemplo: Proceso de cambiar de marcha en el coche.

De forma esquemática, las etapas para resolver un problema son:



```
Algoritmo titulo_algoritmo
    acción 1;
    acción 2;
    .
    .
    .
    acción n;
FinAlgoritmo
```


4. Algoritmos

Los algoritmos tienen que reunir las siguientes características:

- 1) **Precisión:** Los pasos a seguir en el algoritmo deben de ser precisados claramente.
- 2) **Determinismo:** Dado un conjunto de datos idénticos de entrada siempre debe arrojar los mismos resultados.
- 3) **Finitud:** El algoritmo, independientemente de la complejidad del mismo, siempre debe tener una longitud finita.

4. Algoritmos

Una de las formas más sencilla y natural de escribir un algoritmo, es usando pseudocódigos.

Es una mezcla entre lenguaje natural y sintaxis de lenguajes de alto nivel, que permite no estar atado a ningún lenguaje en particular.

Ejemplo:

```
// Leer el radio de un círculo y calcular e imprimir su superficie y su circunferencia.
```

```
// Entradas: Radio del círculo (Variable RADIO).
```

```
// Salidas: Superficie del círculo (Variable SUPERFICIE) y Circunferencia del círculo (Variable PERIMETRO)
```

```
// Variables: RADIO, SUPERFICIE, PERIMETRO de tipo REAL
```

Proceso Círculo

Definir radio,superficie,perimetro como Real;

Escribir "Introduce el radio de la circunferencia:";

Leer radio;

$superficie \leftarrow \pi * radio^2$;

$perimetro \leftarrow 2 * \pi * radio$;

Escribir "La superficie es ",superficie;

Escribir "El perímetro es ",perimetro;

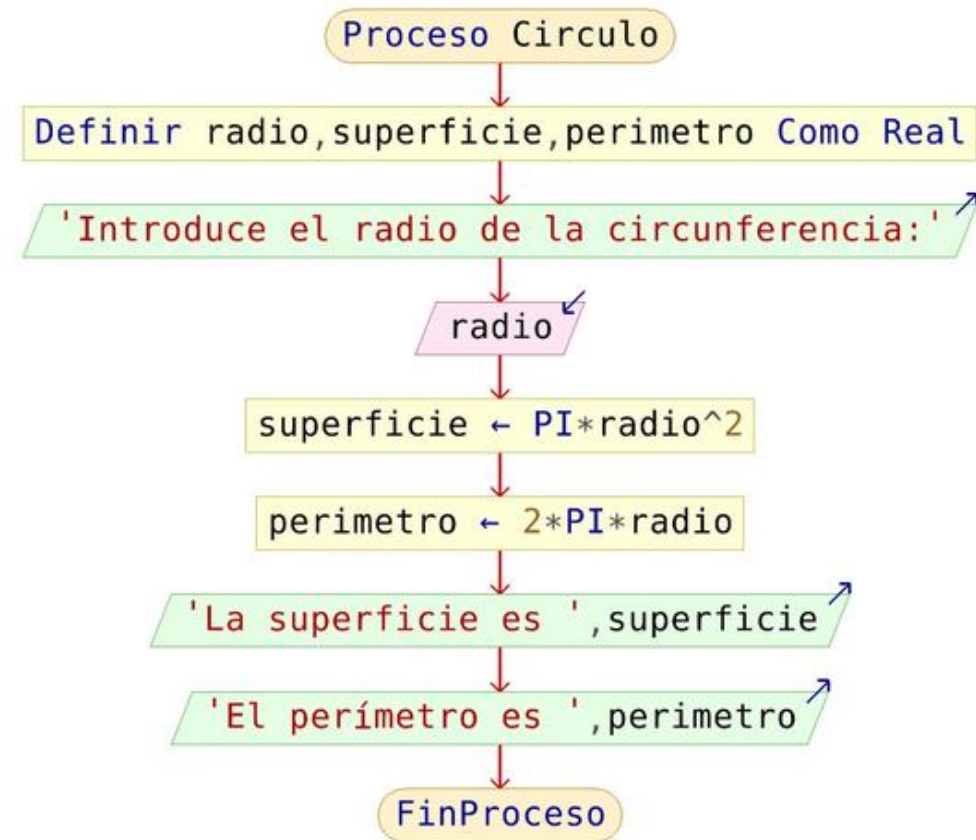
FinProceso

5. Diagramas de flujos

Definición: Un diagrama de flujo es una esquematización gráfica de un algoritmo.

Muestra gráficamente los pasos o procesos a seguir para alcanzar la solución del problema.

Ejemplo, procesamiento de datos:



6. ¿Qué es internet?

Según la Wikipedia:

Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP (Transmission Control Protocol/Internet Protocol o Protocolo de control de transmisión/Protocolo de Internet)., lo cual garantiza que **las redes físicas heterogéneas que la componen formen una red lógica única** de alcance mundial.

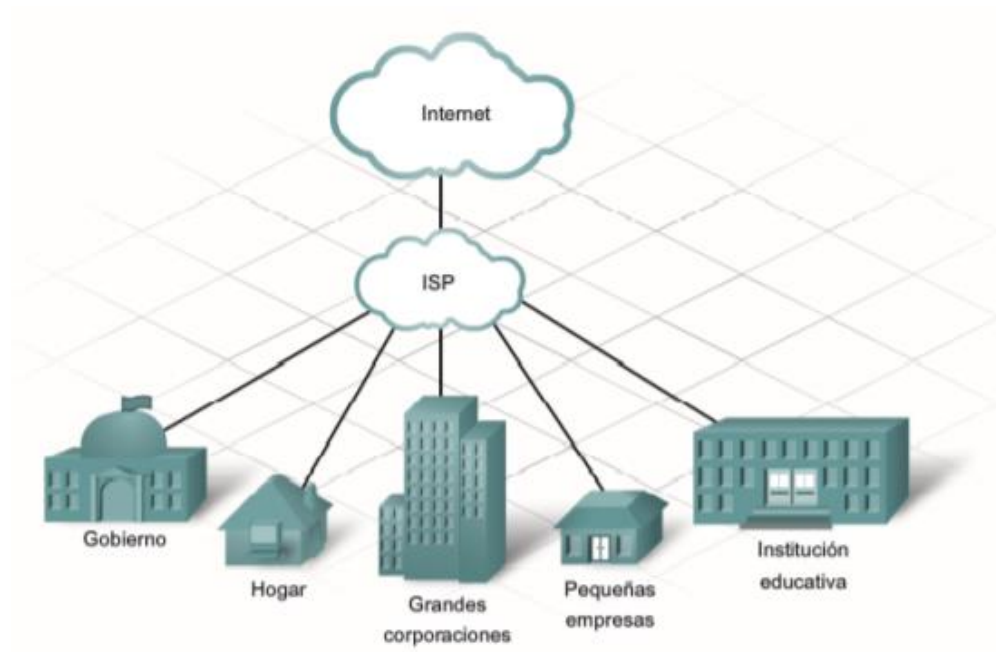
Uno de los servicios que más éxito ha tenido en internet ha sido la **World Wide Web** (WWW o la Web), hasta tal punto que es habitual la confusión entre ambos términos.

La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto y utiliza **internet como medio de transmisión**.



7. ISP

ISP son las siglas de **Internet Service Provider**, es decir, proveedor de servicios de Internet. Para poder conectarnos a internet, lo hacemos mediante la infraestructura que el ISP nos proporciona (Movistar, Orange, Vodafone, etc...).



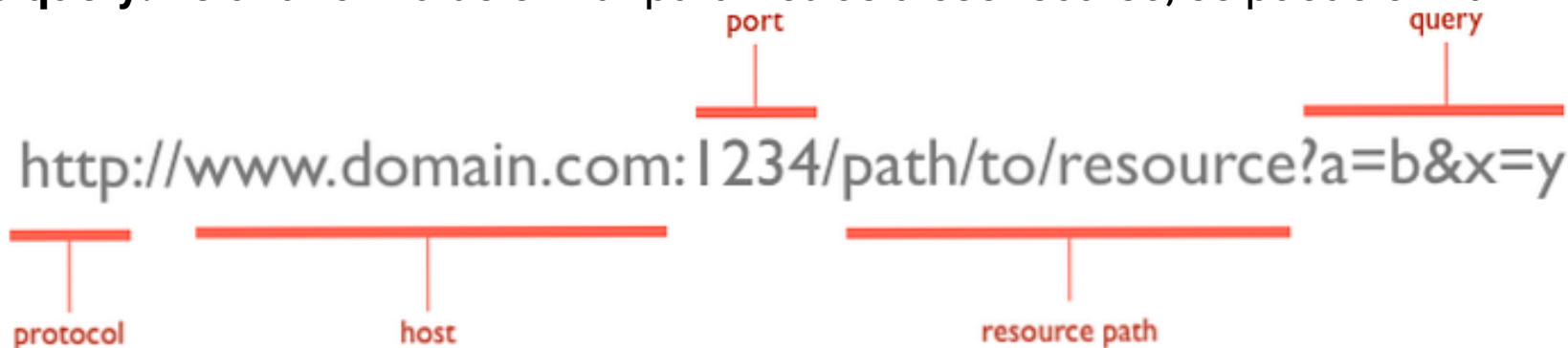
¿ Qué tipo de acceso ofrece un ISP?

- Acceso Teléfono
- Acceso por DSL
- Acceso inalámbrico
- Acceso por satélite
- Fibra óptica

8. URLs

URL (Uniform Resource Locator) es una dirección única de un recurso en la web.
Se compone de varios bloques principales:

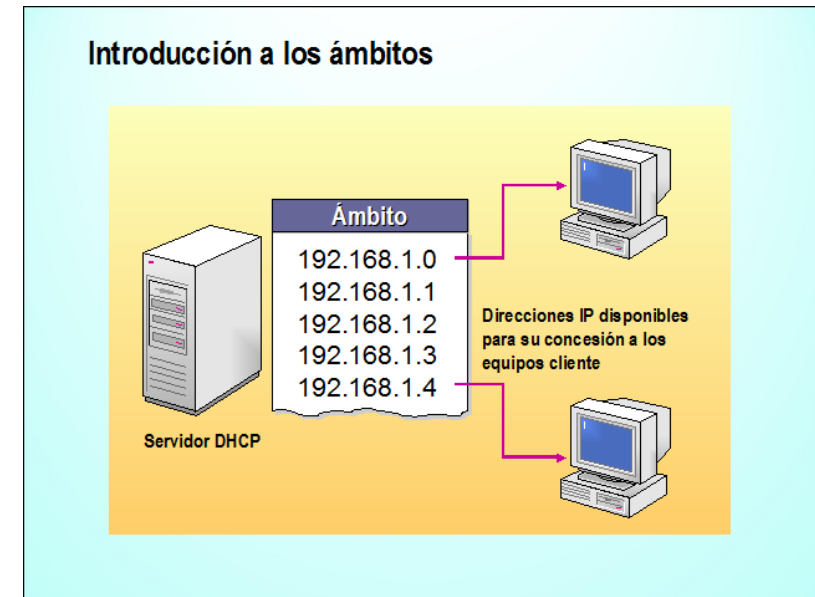
- **Protocolo:** Conjunto de normas que rigen el formato y el control de la interacción entre los diferentes dispositivos.
- **Host:** Es cualquier máquina conectada a una red a través de un dominio y / o una IP definida.
- **Puerto:** Es el punto de entrada y salida de datos de un host que está vinculado a una aplicación específica.
- **Path:** Dentro de la multitud de recursos disponible en un host, la ruta especifica la dirección exacta de forma única.
- **Parámetros o query:** Es una forma de enviar parámetros a ese recurso, se puede enviar información, filtros, etc.



9. IP

La dirección IP es un conjunto de números que identifica, de manera lógica y jerárquica, a una Interfaz en la red (elemento de comunicación/conexión) de un dispositivo (computadora, laptop, teléfono inteligente) que utilice el protocolo (Internet Protocol) o , que corresponde al nivel de red del modelo TCP/IP.

La dirección IP puede cambiar a menudo debido a cambios en la red, o porque el dispositivo encargado dentro de la red de asignar las direcciones IP, decida asignar otra IP, por ejemplo, con el protocolo **DHCP** (Dynamic Host Configuration Protocol).



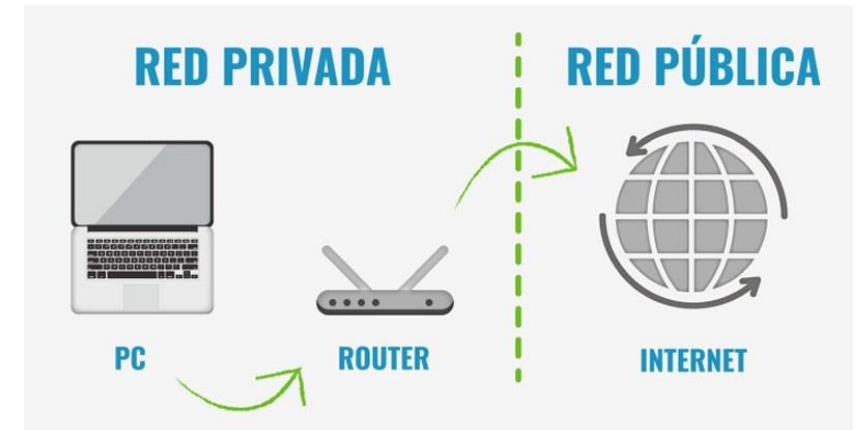
9. IP

Una **dirección IP Privada** se utiliza para identificar equipos o dispositivos dentro de una red doméstica o privada. Se reservan para ello determinados rangos de direcciones:

Clase A: 10.0.0.0 a 10.255.255.255 (Grandes redes – multinacionales)

Clase B: 172.16.0.0 a 172.31.255.255 (Redes medianas – pymes)

Clase C: 192.168.0.0 a 192.168.255.255 (Redes pequeñas – hogares)



En una red local habrá varias direcciones IP Privadas y generalmente una **dirección IP Pública**. Para conectar una red privada con Internet hará falta un “traductor” o NAT (Network Address Translation), que pasará los datos entre las direcciones IP Privadas y las direcciones IP Públicas.

La dirección IP Pública es aquella que ofrece el proveedor de acceso a Internet y se asigna a cualquier equipo o dispositivo conectado de forma directa a Internet. Por ejemplo, los servidores que alojan sitios web, los routers o modems que dan el acceso a Internet.

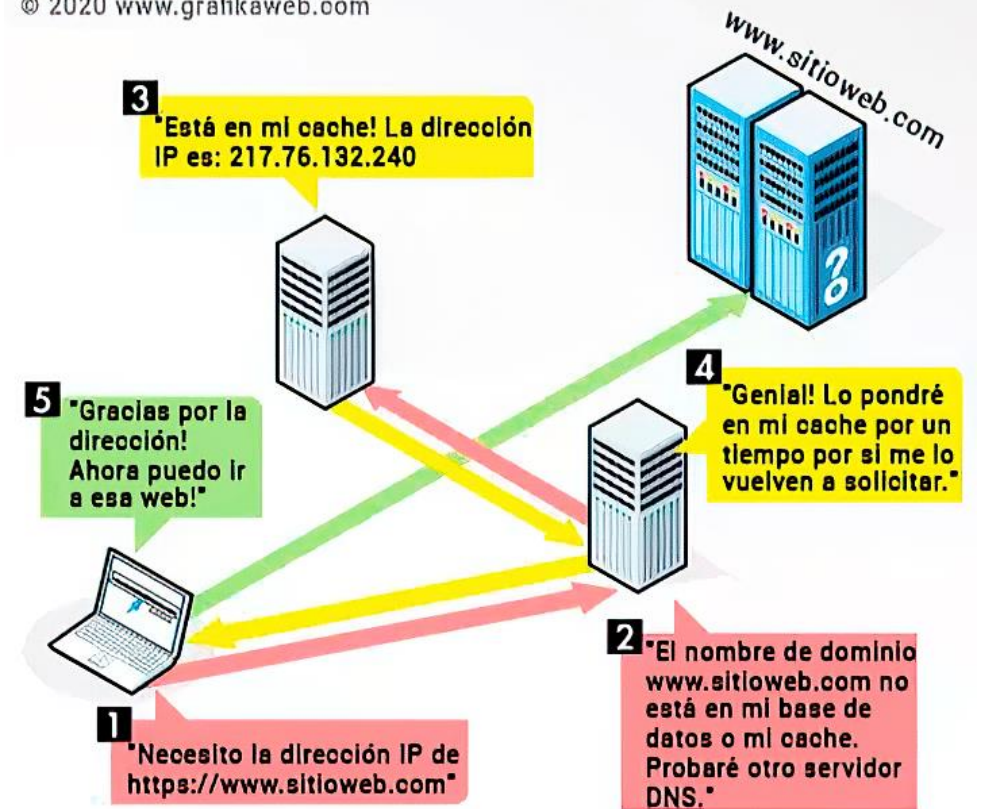
10. Resolución DNS

Todos los recursos de internet están en alguna máquina física con su correspondiente IP.

Por comodidad, no solemos trabajar directamente con estas IPs, sino que se usan los dominios que son más fáciles de recordar.

1. Un equipo pide la resolución de un dominio al servidor DNS
2. El servidor busca en sus registros y en servidores superiores
3. Le devuelve la IP que corresponda a ese dominio
4. El equipo conecta finalmente con el servidor donde están los recursos que necesita

© 2020 www.grafikaweb.com



11. Servidor

El término servidor tiene dos significados:

- El primero hace referencia al ordenador que pone recursos a disposición a través de una red.
- El segundo se refiere al programa que funciona en dicho ordenador.

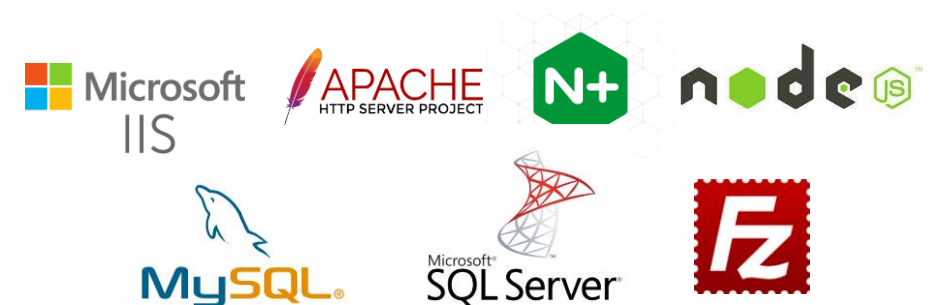
Definición Servidor (hardware):

Un servidor basado en hardware (“host”) es una máquina física integrada en una red informática en la que, además del sistema operativo, funcionan uno o varios servidores basados en software.



Definición Servidor (software):

Un servidor basado en software es un programa que ofrece un servicio especial que otros programas denominados clientes (clients) pueden usar a nivel local o a través de una red.



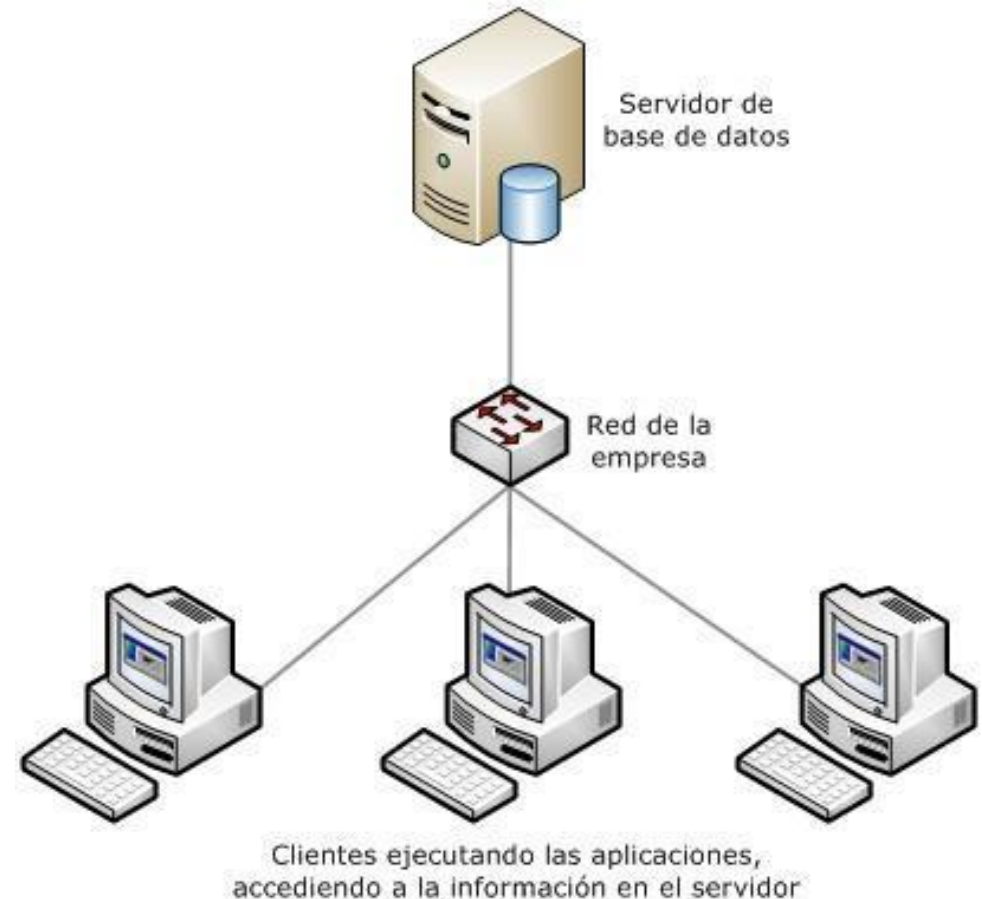
11.2 Modelo Cliente - Servidor

El **cliente es un computador** con una estructura al igual a la que tenemos en nuestras oficinas u hogares que accede a un servidor o a los servicios del mismo a través de Internet o una red interna.

El servidor al igual que el cliente, es una computadora pero con diferencia de que tiene una gran capacidad que le permite almacenar gran cantidad de diversos de archivos, o **correr varias aplicaciones en simultáneo** para así nosotros los clientes poder acceder los servicios.

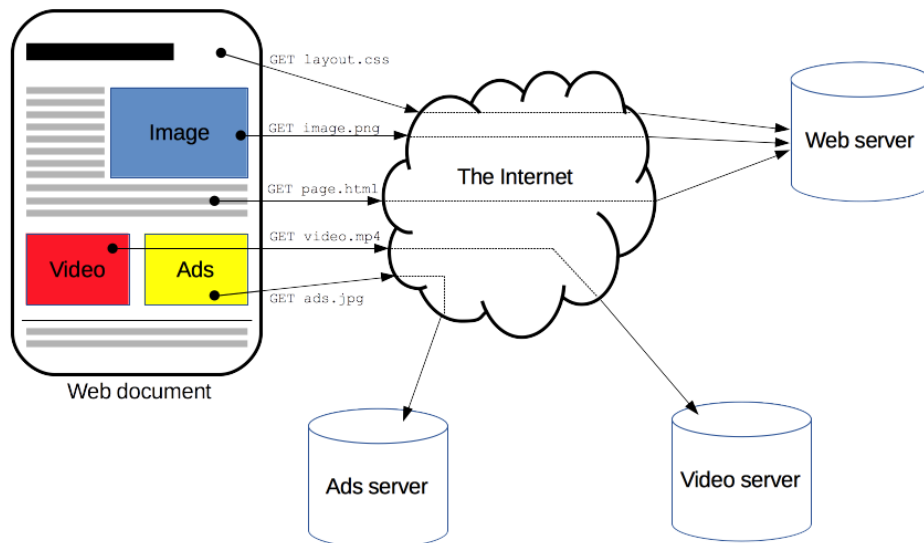
Los servidores pueden ejecutar aplicaciones, sitios web, almacenaje de archivos, diversas bases de datos, entre muchos más.

Es importante mencionar que un cliente también puede tener una función de servidor.



12. Protocolo HTTP

HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo que nos permite realizar una petición de datos y recursos, como pueden ser documentos **HTML**. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura **cliente-servidor**, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Así, una página web completa resulta de la unión de distintos sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc...



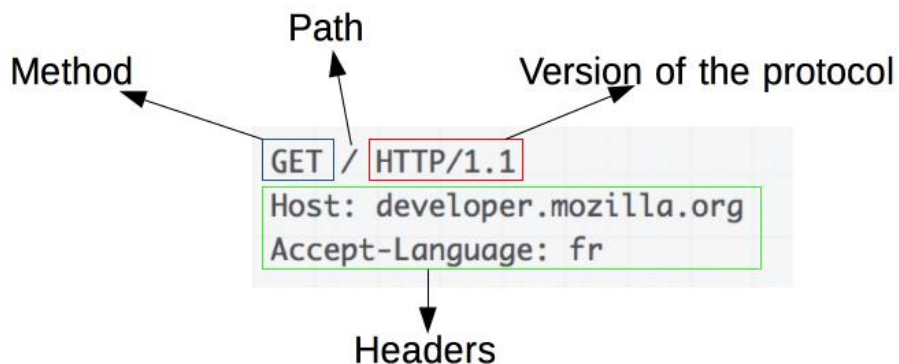
Cientes y servidores se comunican intercambiando mensajes individuales (en contraposición a las comunicaciones que utilizan flujos continuos de datos). **Los mensajes que envía el cliente, normalmente un navegador Web, se llaman peticiones (Request), y los mensajes enviados por el servidor se llaman respuestas (Response).**

12. Protocolo HTTP

Requests

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
Accept: text/html,application/xhtml+xml,..., */*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345
```

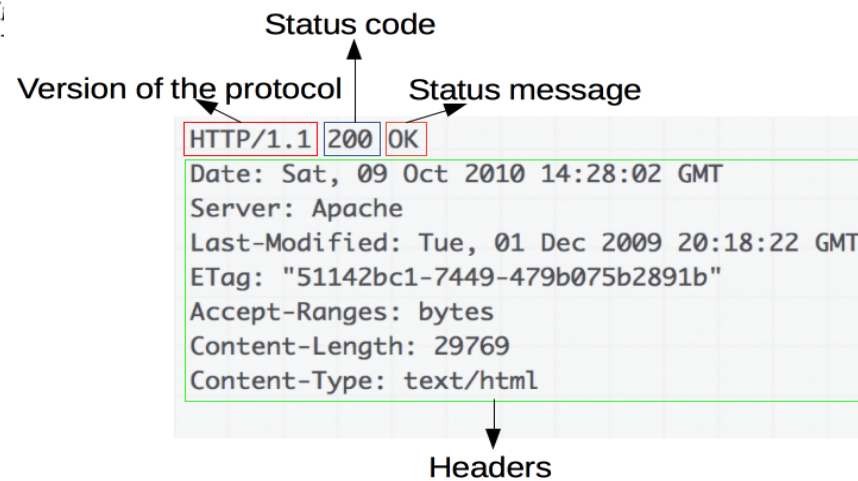
```
-12656974
(more data)
```



Responses

```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
()
```

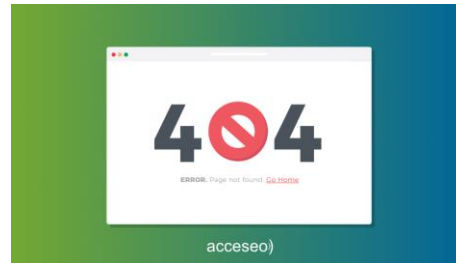
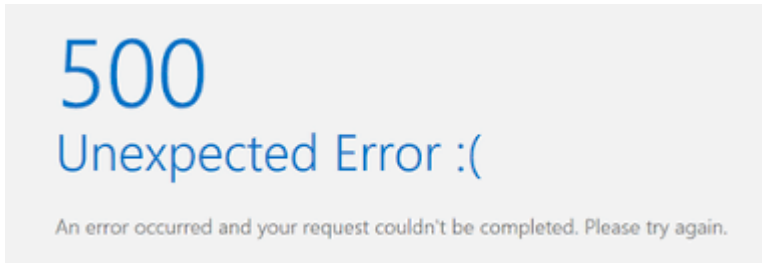


12. Protocolo HTTP – Códigos de Estado

Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica. Las respuestas se agrupan en cinco clases:

- Respuestas informativas (100–199)
- Respuestas satisfactorias (200–299),
- Redirecciones (300–399),
- Errores de los clientes (400–499),
- Errores de los servidores (500–599).

<https://developer.mozilla.org/es/docs/Web/HTTP/Status>



12. Protocolo HTTP – Métodos

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos.

GET

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET únicamente deben recuperar datos.

POST

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE

El método DELETE borra un recurso en específico.

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

12. Protocolo HTTP – Cabeceras / Headers

Cuando visitas cualquier sitio web, tu navegador envía una solicitud al servidor web para obtener datos o información del mismo, por ejemplo, un archivo HTML (es decir, una página web). Tanto en la solicitud, la HTTP-Request, como en la respuesta del servidor, se intercambian, además de datos reales, "información meta". Esta información se resume en la cabecera HTTP.

```
Request URI: http://www.example.com
```

```
HTTP/1.1 200 OK
Content-Encoding: gzip
Age: 521648
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 06 Mar 2020 17:36:11 GMT
Etag: "3147526947+gzip"
Expires: Fri, 13 Mar 2020 17:36:11 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (dcb/7EC9)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 648
```

Cada una de estas líneas recibe el nombre de campo de cabecera (*header field*). Cada una de ellas, excepto la primera, consta de nombres y valores separados por dos puntos.

Request URI: <http://www.example.com>

HTTP/1.1 es la versión válida del protocolo HTTP.

200 OK es el código de estado. Indica que el servidor ha recibido, entendido y aceptado la solicitud.

Content-Encoding y **Content-Type** proporcionan información sobre el tipo de archivo.

Age, **Cache-Control**, **Expires**, **Vary** y **X-Cache** se refieren al *caching* del archivo.

Etag y **Last-Modified** se utilizan para el control del archivo entregado.

Server se refiere al software del servidor web.

Content-length es el tamaño del archivo en bytes.

13. Front-End vs Back-End



FRONT END



BACK END

Módulo 2 - Tema 0

Introducción al Desarrollo Web



**CODE
SPACE**
ACADEMY