

UD1_Tareas

Despliegue de aplicaciones web

DAW2

IES Los Sauces

Fecha

Manuel Martin Alonso

David Aparicio Sir

Josué Martínez Fernández

Luis Pérez Astorga

Ricardo Santiago Tomé

Alejandro Otálvaro Marulanda

Contenido

1. Especificaciones técnicas de tu ordenador personal (Procesador, memoria, discos, sistema operativo, ...)	3
2. Estudio detallado del protocolo HTTP y HTTPS	3
3. ¿Qué navegadores son los más utilizados en el mundo? ¿Y qué sistemas operativos?	6
4. Navegadores: compatibilidad con HTML5, usando la página HTML5test - How well does your browser support HTML5?	7
5. ¿Qué es un motor de navegación web? Listado de Motores de navegador web con información general sobre ellos (desarrolladores, tipo de licencia, lenguaje que se ha programado...)	7
6. ¿Qué es un CMS? ¿Cuáles son los CMS más utilizados?	7
7. ¿Qué tecnologías se utilizan en cada una de las capas de la arquitectura web?	8
8. Existen multitud de servidores web en el mercado. La empresa Netcraft se encarga de hacer estadísticas y mediciones sobre la utilización de los distintos servidores web. Se pide una gráfica con la utilización de los principales servidores web y las principales características de los cinco primeros.	9
9. Diferencias y semejanzas entre Saas, PaaS e IaaS	10
Apache Tomcat 9.X	10
Apache Tomcat 8.X	10
Apache Tomcat 7.X	11
11. URI, URN Y URL: definición, sintaxis (estructura) y diferentes ejemplos	11
12. Escalabilidad horizontal: Ventajas e inconvenientes	12
Ventajas:	12
Desventajas:	12
13. Escalabilidad vertical: Ventajas e inconvenientes.	12
Ventajas:	12
Desventajas:	13

1. Especificaciones técnicas de tu ordenador personal (Procesador, memoria, discos, sistema operativo, ...)

Procesador	I5-7400
Memoria	16Gb
Disco duro	SSD 250GB HDD 1TB
Placa Base	Asus H110M-R
Sistema Operativos	Windows 10 Pro
Software	NetBeans 15 Eclipse 2021 Visual Studio Code 1.72 Filezilla Libre Office Opera Firefox Chrome

2. Estudio detallado del protocolo HTTP y HTTPS.

HTTP

El protocolo HTTP(Protocolo de transferencia de hipertexto) fue desarrollado en 1990 y actualmente es el mas usado en internet.

Este protocolo se ubica en la capa de aplicación y utiliza el protocolo de transporte TCP. El protocolo HTTP utiliza el puerto 80.

Versiones del HTTP:

0.9	Lanzada en 1991 y solo soporto el comando GET
1.0	Lanzada en 1996, actual mente sigue en uso y soporta los métodos de petición GET, HEAD, POST. Añadieron la cabecera que nos permiten transmitir meta-data y también nos permite transmitir mas documentos además de HTML.
1.1	Lanzada en 1999, en esta versión por defecto están activas las conexiones persistentes. Permite al cliente enviar múltiples peticiones.
2.0	Lanzada en 2005, cambio la forma de transferencia a datos de texto plano a binario(Haciendo que no se pueda leer directamente), gracias a este cambio se pudo optimizar.
3.0	Lanzado en 2018, cambiaron de TCP a UDP para el protocolo de transporte subyacente.

<https://es.ccm.net/contents/264-el-protocolo-http>

Como funciona HTTP:

1º El usuario manda una petición HTTP al servidor mediante el navegador.

2º EL servidor web recibe la petición, procesa la petición y lanza al usuario la pagina web que a pedido

3º El navegador recibe la paganía web y la visualiza

Solicitud HTTP:

- **Una línea de solicitud:** En esta línea especificamos el tipo de documento, solicitado, el método que se aplicara y la versión del protocolo.
- **Encabezado de solicitud:** Un conjunto de líneas opcionales que nos permite añadir información sobre la solicitud, navegador, sistema operativo, hardware que utiliza el cliente.
- **El cuerpo de la solicitud:** Un conjunto de líneas opcionales que están separadas por una línea en blanco. Nos permiten enviar datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

```
GET https://es.ccm.net/ HTTP/1.0 Accept : Text/html If-
Modified-Since : Saturday, 15-January-2000 14:37:11 GMT User-
Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)
```

Métodos de petición HTTP

- **GET:** Solicita una representación de un recurso en específico. No puede enviar datos, solo recuperarlos, la excepción se da cuando dichos datos se envían como un parámetro en la URL que realiza la petición.
- **HEAD:** Es similar al Get pero este no envía datos como un parámetro URL.
- **POST:** Este método puede enviar datos al servidor por medio del cuerpo(body). EL tipo de cuerpo de solicitud se define en la cabecera Content-Type.
- **PUT:** Crea un nuevo elemento o reemplaza una representación del elemento de destino con los datos de la petición. Este método es Idempotente (Significa que da igual cuantas veces lo llames siempre va a tener el mismo efecto).
- **DELETE:** Nos permite eliminar un recurso específico. Este método es Idempotente (Significa que da igual cuantas veces lo llames siempre va a tener el mismo efecto).
- **CONNECT:** Inicia la comunicación entre el cliente y el servidor, también se utiliza para crear una comunicación de túnel.
- **OPTIONS:** Comprueba que opciones de comunicación posee el recurso de destino
- **TRACE:** Efectúa una prueba del camino que efectúa el recurso.
- **PATCH:** Nos permite hacer modificaciones parciales de un recurso en concreto.

Encabezados HTTP:

Accept	Tipo de contenido aceptado por el navegador(texto/html)
Accept-Charset	Juego de caracteres que el navegador espera
Accept-Encoding	Codificación de datos que el navegador acepta
Accept-Language	Idioma que el navegador espera(Predeterminado es el ingles)
Authorization	Identificación del navegador en el solicitud
Content-Encoding	Tipo de codificación para el cuerpo de la solicitud
Content-Language	Tipo de idioma en el cuerpo de la solicitud
Content-Length	Tamaño del cuerpo de la solicitud
Content-Type	Tipo de contenido del cuerpo de la solicitud
Date	Fecha en que comienza la transferencia de datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
From	Permite especificar la dirección de correo electrónico del cliente, También permite especificar que debe enviarse el documento si ha sido modificado desde una fecha en particular.
Link	Vínculo entre dos direcciones URL

Orig-URL	Dirección URL donde se origina la solicitud
Referer	Dirección URL desde la cual se realiza la solicitud
User-Agent	Cadena con información sobre el cliente.(nombre, versión del navegador...)

La respuestas HTTP tiene 3 partes:

- **Una línea de estado:** Esta línea especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código.
- **Encabezado de respuesta:** Líneas opcionales que permiten aportar información adicional sobre la respuesta y/o el servidor.
- **El cuerpo de la respuesta:** Contiene el documento solicitado

```
HTTP/1.0 200 OK Date: Sat, 15 Jan 2000 14:37:12 GMT Server :
Microsoft-IIS/2.0 Content-Type : text/HTML Content-Length :
1245 Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT
```

Encabezado de Respuesta HTTP:

Content-Encoding	Tipo de codificación para el cuerpo de la respuesta
Content-Lenguaje	Tipo de idioma del cuerpo
Content-Length	Tamaño del cuerpo de la respuesta
Date	Fecha en que comienza la transferencia
Expires	Fecha límite de uso de los datos
Forwarded	Se utiliza para quipos intermediarios entre el navegador y el servidor(proxi)
Location	Redireccionamiento a una nueva dirección URL
Server	Características del servidor que envió la respuesta

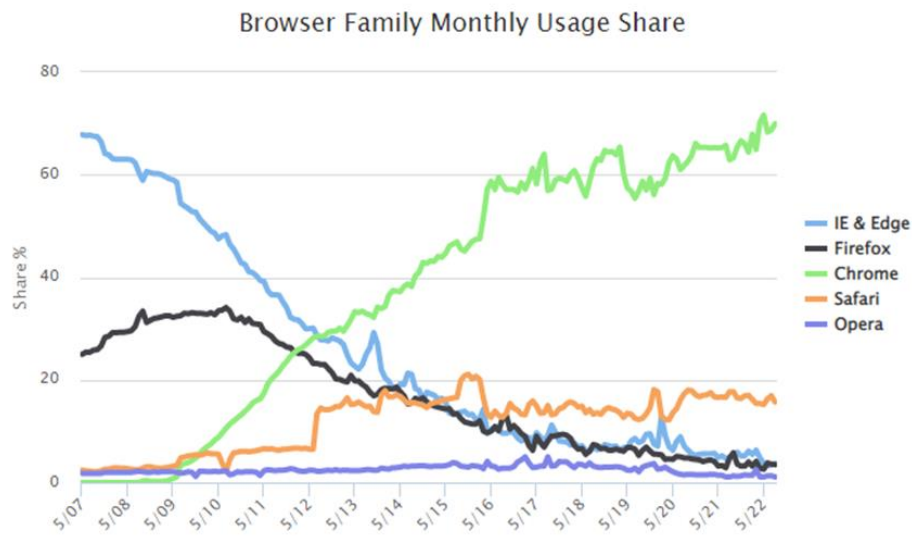
Codigos de respuesta:

10X	Mensaje de información	Este código de error no está presente en la Versión 1.0 de HTTP
20X	Ejecución correcta	Mensaje de correcta ejecución
30x	Redirección	Este código indica que el recurso a sido cambiado de ubicación
40X	Error debido al cliente	Este código indica que la solicitud es incorrecta
50X	Error debido al servidor	Código que indican que existe un error interno en el servidor

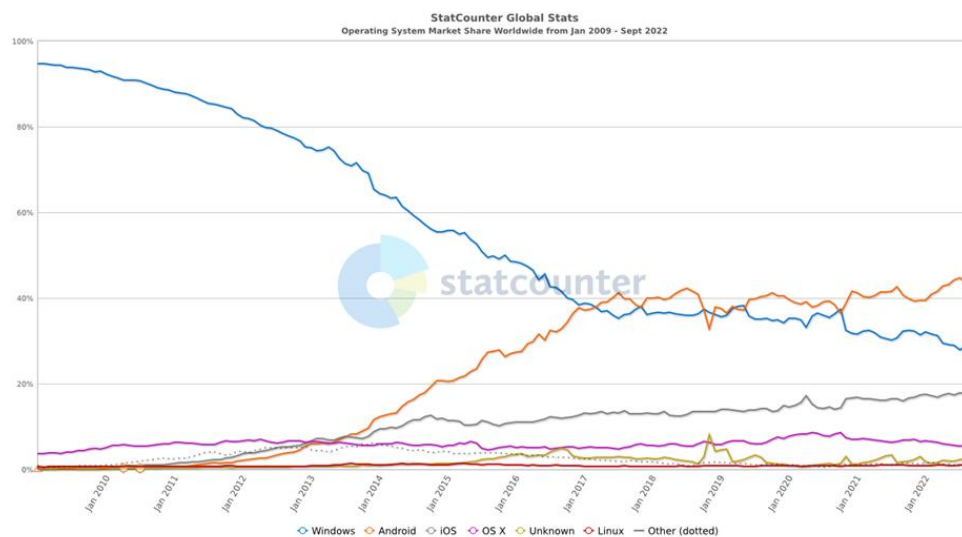
HTTPS

Es una extensión del protocolo HTTP, este en vez de enviar los datos sin cifrar el protocolo HTTPS que utiliza SSL (Cifrado con certificados digitales) y TLS (Protocolo de la capa de transporte, que se usa para transferir de un equipo a otro de forma segura)

3. ¿Qué navegadores son los más utilizados en el mundo? ¿Y qué sistemas operativos?



<https://www.w3counter.com/trends>



<https://gs.statcounter.com/os-market-share#monthly-200901-202209>

4. Navegadores: compatibilidad con HTML5, usando la página HTML5test - How well does your browser support HTML5?

OVERVIEW

	Chrome	Opera	Firefox	Edge	Safari
Upcoming	67 528		60 497	18 496	11.2 477
Current	66 528	45 518	59 491	17 492	11.1 471
Older	65 528	37 489	58 486	16 476	11 452
	64 528	30 479	57 486	15 473	10.1 406
	63 528	12.10 309	56 478	14 460	10.0 383
	62 528		55 478	13 433	9.1 370
	61 526		54 474	12 377	9.0 360
	60 523		53 474	Internet Explorer	8.0 354
				11 312	

<https://html5test.com/results/desktop.html>

5. ¿Qué es un motor de navegación web? Listado de Motores de navegador web con información general sobre ellos (desarrolladores, tipo de licencia, lenguaje que se ha programado...).

Un **motor de búsqueda** o **buscador** es un sistema informático que [busca archivos](#) almacenados en [servidores web](#) gracias a su araña web.

Google Desarrollador Google licencia lenguaje C++, Python y JavaScript

Baidu Desarrollador Robin Li y Eric Xu licencia lenguaje

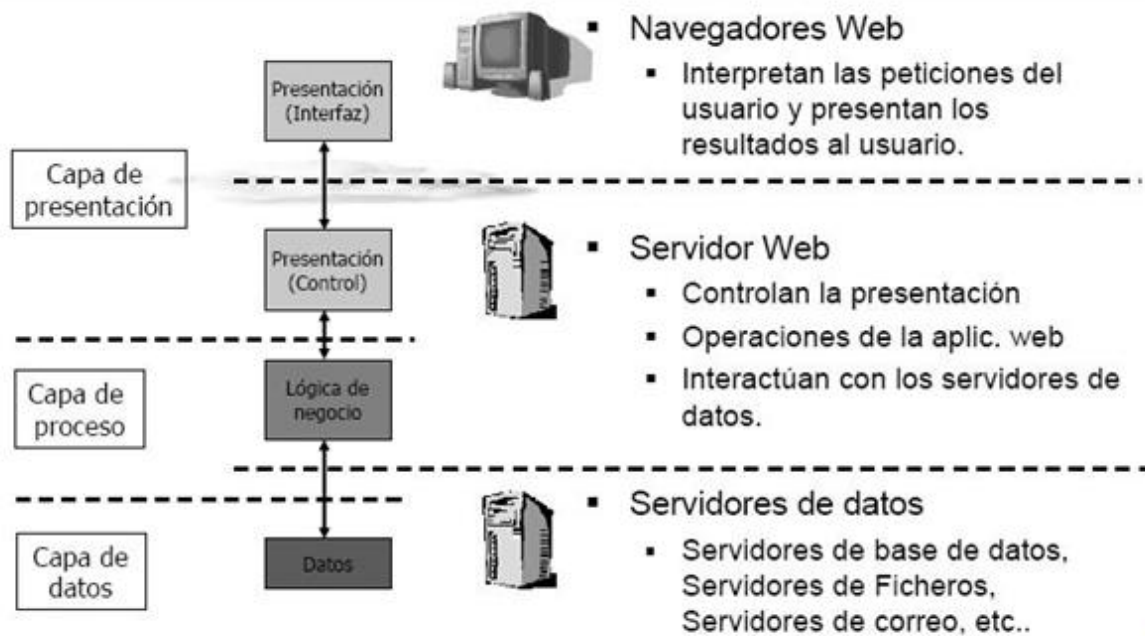
Bing Desarrollador Microsoft licencia lenguaje ASP.NET

6. ¿Qué es un CMS? ¿Cuáles son los CMS más utilizados?

El término CMS proviene del inglés Content Management System, que significa **Sistema de Gestión de Contenidos**. Es un sistema online que nos permite poner en marcha un sitio web de forma práctica y rápida.

WordPress, Joomla, prestashop

7. ¿Qué tecnologías se utilizan en cada una de las capas de la arquitectura web?



En la capa de presentación se utilizan las tecnologías de los navegadores, para interactuar de forma gráfica con la aplicación.

Javascript, html, css, lenguajes de marcado...

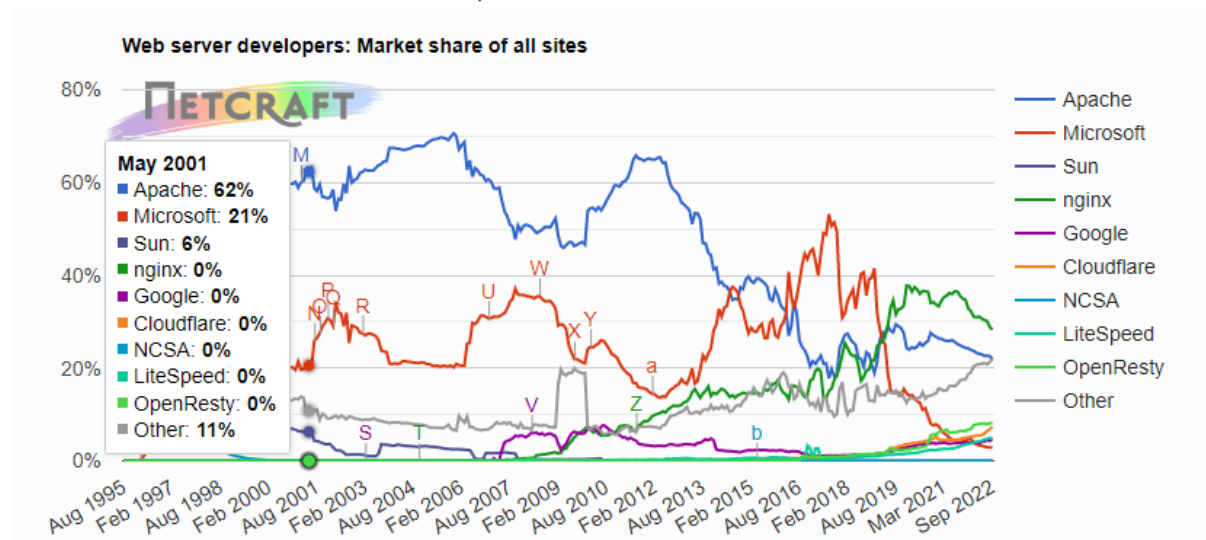
En la capa de proceso Se encuentran los softwares intermedios, así como la lógica de negocio. En esta capa se encuentra el grueso de la aplicación y todas aquellas tecnologías encargadas de trabajar con los datos para posteriormente enviárselos a la capa de presentación.

Php, u otros lenguajes de programación, servidores y servicios web, dns, dhcp...

En la capa de datos se encuentran aquellas tecnologías relacionadas con la gestión de los datos, un ejemplo de esto serian las BBDD.

Mysql, MariaDB, MongoDB...

8. Existen multitud de servidores web en el mercado. La empresa Netcraft se encarga de hacer estadísticas y mediciones sobre la utilización de los distintos servidores web. Se pide una gráfica con la utilización de los principales servidores web y las principales características de los cinco primeros.



Servidor apache: Es código abierto, es además software gratuito, y multiplataforma.

Servidor Microsoft: Permite el procesamiento y despacho de páginas desarrolladas en tecnología ASP / ASP.NET, aunque también vale aclarar que sirve para interpretar páginas programadas en Perl o PHP.

No es sólo un servidor web, sino también una suite de servicios para la web, ya que ofrece también servicios de SMTP y FTP por ejemplo. Hoy se integra naturalmente con Microsoft Azure.

Servidor Sun: También conocido como SJSAS, es un tipo de servidor web con licencia BSD que está enfocado en brindar despacho a aplicaciones hechas en Java.

Soporta integración nativa con Sun Java Studio Enterprise, Sun Java Studio Creator y NetBeans.

Otras características incluyen soporte para PHP, WebDAV, filtros NSAPI, JDBC y compresión HTTP.

Servidor Nginx: Open source y gratuito (aunque también existe una versión comercial) destaca por su alto rendimiento. Incluye además funciones como servidor proxy reverso HTTP, balanceador de carga, así como POP3 y IMAP

Servidor Google: Google Web Server (abreviado como GWS), no se puede descargar desde ningún lugar, ya que no está disponible para el público.

Se trata de un servidor web privado escrito en C++, que es utilizado por Google para la mayoría de su infraestructura web.

9. Diferencias y semejanzas entre Saas, PaaS e IaaS

SaaS Servicio basado en la web los usuarios acceden sin prestar atención al software

PaaS (Plataforma como servicio) se presenta como la alternativa para aquellos desarrolladores que solo quieren preocuparse de construir la aplicación. La infraestructura la proporciona la plataforma y se ocupa tanto de su gestión como de su mantenimiento.

IaaS (Infraestructura como servicio) sistema idóneo para desarrolladores que deseen encargarse de la gestión y administración de su infraestructura. Ofrece más control que PaaS de modo que el desarrollador es el responsable de todo lo relacionado con el mantenimiento de la estructura así como de escalar sus aplicaciones en función de las necesidades.

10. Diferentes versiones disponibles de Apache Tomcat y sus diferentes versiones de las especificaciones

Apache Tomcat 9.X

Apache Tomcat 9. El desarrollo de los tiempos es el enfoque actual. Está construido en Tomcat 8.0. 8.5 x y. X y realice el Servlet 4.0, JSP 2.3, EL 3.0, STAX 1.1 y JASPI 1.1 Especificaciones (versión requerida Java EE Platform). Además, incluye las siguientes mejoras importantes:

- Añadido http / 2
- Se agregó soporte de TLS utilizando OpenSSL JSSE Conector (NIO y NIO2)
- Soporte adicional TLS Host virtual (SNI)

Apache Tomcat 8.X

Apache Tomcat 8.0.x está construido en Tomcat 7.0. X y realice el servlet 3.1, JSP 2.3, EL 3.0 y WebSocket 1.1 Especificación. Además, incluye las siguientes mejoras importantes:

- Una sola implementación de recursos comunes reemplaza varias versiones anteriores de recursos para proporcionar características de extensión.

Apache Tomcat 8.5.x admite el mismo servlet, JSP EL como la versión de Especificación de Apache Tomcat 8.0.x. Además, también implementa la especificación JASPI 1.1.

Fue establecido en marzo de 2016 desde Tomcat 9.0.0. Versión M4 (α) Milestone. Proporciona soporte HTTP / 2 y otras características Tomcat 9. X Base de código, mientras se ejecuta con Tomcat 8.0 Tiempo de ejecución y compatibilidad. (No se puede crear una versión estable de Tomcat 9.0, el Tomcat 9 de la especificación Java EE es, en última instancia, solo unos años más tarde.

Tomcat 8.5 se considera que reemplaza a Tomcat 8.0.

Apache Tomcat 8.5. X incluye las siguientes mejoras importantes:

- Soporte http / 2 (es necesario que se agregue [Gato](#) Biblioteca)
- Se agregó soporte de soporte TLS utilizando OpenSSL JSSE Conector (NIO y NIO2)
- Soporte adicional TLS Host virtual (SNI)

La siguiente técnica se elimina en Apache Tomcat 8.5.x:

- Implementación biológica HTTP y conector americano
- Apoyy API COMET

Hay un cambio importante en muchos campos en el capó, lo que resulta en un mejor rendimiento, estabilidad y costo general. Consulte el registro de actualización de Apache Tomcat 8.5 para obtener más información.

Tomcat 8.0 debe estar al tanto de Tomcat 8.0 Los usuarios han alcanzado [El fin de la vida](#). Tomcat 8.0 usuarios. X debe actualizarse a Tomcat 8.5. X o superior.

Apache Tomcat 7.X

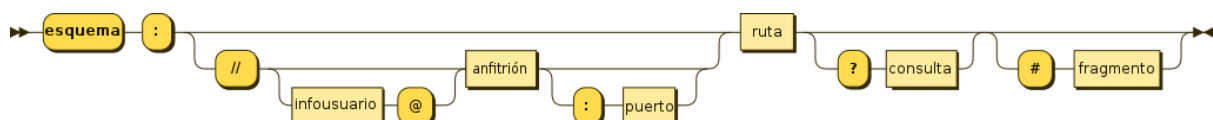
Apache Tomcat 7. El establecimiento se mejora en Tomcat 6.0. X e implemente el Servlet 3.0, JSP 2.2, ER 2.2 y WebSocket 1.1 Especificación. Además, incluye las siguientes mejoras:

- Memoria de aplicación web Detección y prevención de fugas.
- Aplicación mejorada de administrador de seguridad y administrador de host
- Protección universal de CSRF
- El soporte incluye contenido externo directamente en la aplicación web
- Reconstrucción (conector, ciclo de vida) y una gran cantidad de limpieza de código interno

Información recogida en: <https://programmerclick.com/article/90222249955/>

11. URI, URN Y URL: definición, sintaxis (estructura) y diferentes ejemplos.

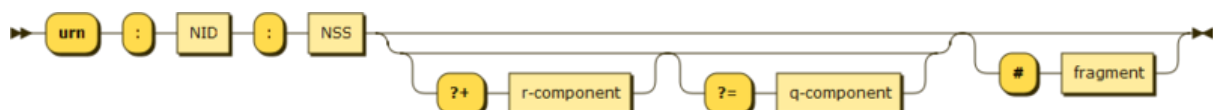
URI es una cadena de caracteres que identifica los recursos –físicos o abstractos– de una red de forma unívoca.



URN Los URN identifican recursos en la [web](#), pero a diferencia de los URL, no indican exactamente dónde se encuentra ese objeto.

Básicamente, un *Uniform Resource Identifier (URI)*, está constituido ya bien sea por un [URL](#) más un [URN](#) o por solo uno de ellos:

Ejemplos https://en.wikipedia.org/wiki/Uniform_Resource_Name#Examples



URL es un identificador de recursos uniforme (*Uniform Resource Identifier*, **URI**) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo. Están formados por una secuencia de caracteres de acuerdo con un formato modélico y estándar que designa recursos en una red como, por ejemplo, Internet.

:esquema://máquina/directorio/archivo

<https://www.wikipedia.org/>

12. Escalabilidad horizontal: Ventajas e inconvenientes

El escalado horizontal se consigue aumentando el número de servidores que atienden una aplicación. Para ello, un grupo de distintos servidores se configura para atender las peticiones de manera conjunta (es lo que se denomina cluster) y la carga de trabajo se distribuye entre ellos a través de un balanceador. Cada uno de esos servidores se conoce como nodo y el escalado se realiza simplemente agregando un nuevo nodo al cluster.

Este escalado es bastante más potente, pero sin embargo requiere una mayor configuración para poder realizarse, no solamente para crear la red de servidores de un cluster, sino también realizando una arquitectura de aplicación, a nivel de software, capaz de adaptarse a este tipo de funcionamiento.

Ventajas:

- El escalado es prácticamente infinito.
- Permite alta disponibilidad.
- Permite un correcto balanceo de carga entre los servidores.

Desventajas:

- Requiere mayor configuración, que puede llegar a ser difícil de realizar.
- Necesita que la aplicación esté construida de modo que soporte escalamiento vertical.
- Aunque más potente y de mejor rendimiento, suele ser una opción menos económica, ya que requiere de varios servidores.

13. Escalabilidad vertical: Ventajas e inconvenientes.

El escalado vertical tiene mucho que ver con el hardware del servidor de la aplicación. Se consigue de una manera muy sencilla: aumentando los recursos del servidor. Principalmente, en lo que respecta a la capacidad de procesamiento, memoria y almacenamiento. Este tipo de escalado es bastante sencillo de alcanzar, ya que únicamente requiere una intervención en el hardware del equipo, aumentando los recursos o incluso cambiando completamente de servidor. Sin embargo, el beneficio que se puede llegar a obtener también es limitado.

Ventajas:

- Facilidad de implementación y configuración.
- No requiere un diseño específico en la aplicación y su arquitectura para funcionar.
- Puede ser más económico.

Desventajas:

- Está limitado a la capacidad de un único servidor.
- No aporta beneficios en relación a la alta disponibilidad.