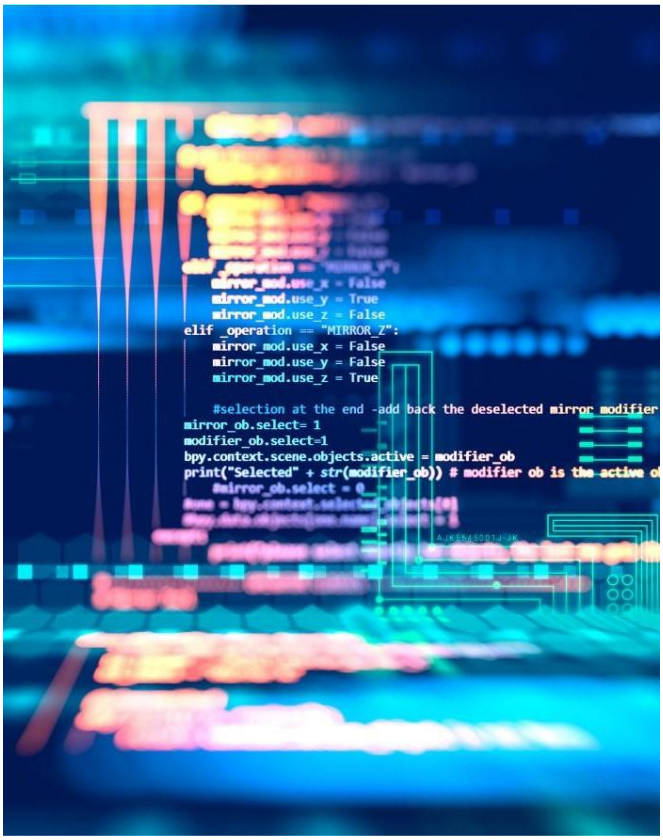


IMPLANTACIÓN DE APLICACIONES WEB

Despliegue de aplicaciones web (DAW)

2º DESARROLLO DE APLICACIONES WEB
30/09/2020

BEATRIZ MERINO MACÍA

Contenido

INTRODUCCIÓN

En este documento realizaremos un estudio sobre la **Implantación de Aplicaciones Web**, basándonos en el contenido de la Unidad 1.

Este estudio se llevará a cabo mediante la respuesta de una serie de preguntas que abarcan todo el contenido necesario para trabajar la Unidad 1.

DESARROLLO DE LOS CONTENIDOS

1. Especificaciones técnicas de tu ordenador personal

Elemento	Valor
Versión	10.0.18363 compilación 18363
Nombre del SO	Microsoft Windows 10 Home
Fabricante del SO	Microsoft Corporation
Fabricante del sistema	TOSHIBA
Modelo del sistema	SATELLITE C55D-A-14E
Tipo de sistema	SO de 64 bits procesador basado en x64
Procesador	AMD E1-2100 APU with Radeon(TM) HD Graphics, 1000 Mhz, 2 procesadores principales, 2 procesadores lógicos
Modo de BIOS	Heredado
Directorio de Windows	C:\WINDOWS
Directorio del sistema	C:\WINDOWS\system32
Configuración regional	España
Nombre de usuario	BEA\Usuario
Zona horaria	Hora de verano romance
Memoria física instalada (RAM)	4,0 GB
Memoria virtual total	6,08 GB

2. Estudio detallado del protocolo HTTP.

Protocolo de Transferencia de HiperTexto (**HTTP**), un sencillo **protocolo cliente-servidor** que articula los intercambios de información entre los clientes Web y los servidores HTTP.

La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: **un servidor escucha en un puerto de comunicaciones TCP, y espera las solicitudes de conexión de los clientes Web**.

Una vez que se establece la conexión, **el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.**

Un **cliente** establece una conexión con un servidor y **envía** un mensaje con los datos de **la solicitud**.

El **servidor responde con un** mensaje similar, que contiene el estado de la operación y su posible **resultado**.

Algunos de sus **métodos** más utilizados son:

- El método **GET** se emplea para **adquirir un recurso** ubicado en un servidor web, que **devuelve los metadatos y el recurso** que se precisa.
- El método **HEAD** realiza una función similar a la anterior, pero **sólo se solicitan los metadatos** y no el recurso en sí.
- El método **POST** se usa para **enviar información** a un recurso web del servidor, **no para crear un elemento nuevo**.
- El método **PUT crea un recurso nuevo** en el servidor **o** si éste ya existe **lo reemplaza**.
- El método **DELETE** solicita **borrar el recurso especificado**.
- El método **TRACE** sirve para el **diagnóstico de errores o detección de servidores intermedios** en la conexión mediante la monitorización de mensajes.
- El método **CONNECT** solicita **saber si se tiene acceso a un host**.
- El método **PATCH** solicita **sobrescribir completamente un recurso**. Se utiliza para **actualizar, de manera parcial, una o varias partes**.

3. Se pide buscar en Internet diferentes aplicaciones web y clasificarlas según su tipo, como mínimo dos páginas de cada tipo.

Aplicación web estática: son las **más sencillas** y no suelen tener muchos cambios al no ser sencillos. Están desarrolladas con **código HTML y CSS** y pudiendo mostrar algún banners o vídeos, entre otros. La razón para no tener muchas variaciones es la descarga del HTML, modificado y subida del mismo.

Ejemplo:
- Página de Error 404
- Web Quest, El principal objetivo reside en aprender los contenidos del tema elegido de una manera interactiva.

Aplicación web dinámica: son mucho más complejas que las anteriores técnicamente hablando, **actualizando información y contenido cada entrada de un usuario a la web**. Existen muchos **lenguajes** de programación, pero los **más comunes son PHP y JavaScript**. Además de actualizar también se puede **modificar el diseño** de la web.

Ejemplo:
- https://www.google.es/
- https://es.wikipedia.org/wiki/Wikipedia:Portada

Tienda virtual o comercio electrónico: son **conocidas como e-commerce** y están **pensadas para vender productos**. Son **más complejas porque tienen que incluir el método de pago y estar sincronizadas con el stock de la compañía y con la logística**.

Ejemplo:
- https://www.amazon.es/
- https://www.fnac.es/

Portal web app: este tipo **incluyen diferentes categorías y secciones**. **Pueden tener chats, foros o buscador**, entre otras opciones.

Ejemplo:
- https://www.infojobs.net/
- https://www.forocoches.com/

Aplicación web animada: estaban **relacionadas con la tecnología FLASH**, aunque hoy en día lo están con CSS y SVG. Permiten efectos animados. Son muy útiles para diseñadores y desarrolladores, pero, como contrapartida, presentan un problema: no son útiles para mejorar el posicionamiento ni el SEO porque **los buscadores no leen correctamente su información**.

Ejemplo:
- https://www.android.com/
- https://www.youtube.com/?gl=ES&hl=es

Aplicación web con Gestor de contenido: es un software que **ayuda a los usuarios a crear, administrar y modificar contenido en un sitio web sin la necesidad de conocimientos técnicos especializados**.

Se compone de dos **partes** principales:

- **Una Aplicación de Gestión de Contenido (CMA)** – esta es la parte que le permite **agregar y administrar el contenido** de su sitio.
- **Una Aplicación de Entrega de Contenido (CDA)** – este es el proceso de **backend** y detrás de escena que **toma el contenido** que ingresa en el CMA, **lo almacena** correctamente **y lo hace visible** para sus visitantes.

Ejemplos:
- https://es.wordpress.com/
- https://www.joomla.org/

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

Es un tipo de desarrollo software donde **se persigue la separación de las partes de las que está compuesto un sistema** software en capas, éstas **sólo pueden comunicarse con la capa anterior, de la que reciben información y con la capa posterior, a la que envían información.**

Presentación: En esta capa se crea la **interfaz del usuario**. Su única función es pasarle las acciones que realice el usuario a la capa de negocio.

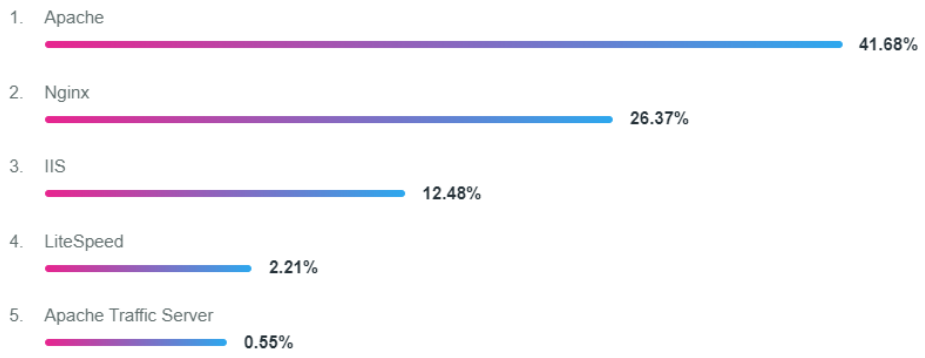
Negocio: En esta capa se gestiona la **lógica de la aplicación**. Es donde se dice qué se hace con los datos. Estará **conectada con la capa de persistencia** para poder realizar sus funciones.

Persistencia: Esta capa se encarga de guardar los datos. Será donde se **gestione todo lo relativo a la base de datos** y a la creación, edición y borrado de datos de ésta.

[1]

[1] Esquema de la Arquitectura de las Aplicaciones Web

5. 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.



[2]

Apache: El web server de referencia para Internet. Nació en 1996 y sigue vigente. **El más usado.**

Ventajas:

- Código abierto
- Software gratuito
- Multiplataforma (Windows, Linux y Unix).

Desventajas:

- Bajo rendimiento cuando se reciben miles de requests.

Nginx: Servidor web de **código abierto y gratuito** (también existe una versión comercial), destaca por su **alto rendimiento**. Está diseñado para ofrecer un bajo uso de memoria y alta concurrencia.

Ventajas:

- Software multiplataforma.
- Consume menos recursos que la mayoría de los servicios con su misma función.
- Alto rendimiento soportando mayor carga y respondiendo.
- Puede ser usado como proxy inverso.
- Podemos integrarlo con Apache, de forma que Nginx procese contenido estático y Apache contenido dinámico.

Desventajas:

- No se integra con PHP de forma nativa. Es necesario usar FastCGI.

[2] Grafica analítica de los servidores web más usados -- https://es.hostadvice.com/marketshare/server/

Microsoft IIS: Internet Information Services, conocido como IIS, es un tipo de servidor web **creado por Microsoft específicamente para** su plataforma de **sistemas operativos Windows**.

Ventajas:

Proporcionan capacidades de servidor web integrado.

Es confiable, seguro y administrable en internet.

Desarrolla y es compatible con las aplicaciones beneficiándose con un único entorno de alojamiento de aplicaciones integrado con compatibilidad total.

Al momento de la instalación permite elegir sobre que servidor web va a correr (Apache oIIS).

Desventajas:

Este servidor no es multiplataforma, solo funciona bajo Windows.

Posee limitaciones en las versiones que no son de la familia "Server".

Posee vulnerabilidades.

LiteSpeed Web Server (LSWS), es un software de **servidor web propietario**. Desarrollado por LiteSpeed Technologies, empresa privada. El software utiliza el mismo formato de configuración que Apache HTTP Server y es **compatible con la mayoría de las funciones de Apache**. También está disponible una variante de código abierto.

Ventajas:

Acorde a la configuración, algunos módulos y los archivos .htaccess de Apache.

Las migraciones más sencillas.

Provee un panel de control propio.

Implementa su propia instalación de un servidor PHP con varias versiones de PHP simultáneamente.

Velocidad y eficiencia.

Desventajas:

Es que es software propietario y su uso requiere de un pago mensual.

Apache Traffic Server

Servidor proxy de almacenamiento en caché rápido, escalable y extensible compatible con HTTP / 1.1 y HTTP / 2.0.

Fue creado por Inktomiy distribuido como un producto comercial llamado Inktomi Traffic Server, antes de que Inktomi fuera adquirido por Yahoo!.

6. Diferencias y semejanzas entre Saas, PaaS e IaaS

IaaS (Infraestructure-as-a-Service) sistema idóneo **para desarrolladores que deseen encargarse de la gestión y administración de su infraestructura**. Mayor control que otras alternativas, el desarrollador es responsable del mantenimiento de la infraestructura, incluso de escalar sus aplicaciones en función de cuáles sean sus necesidades.

Un **ejemplo** es **Amazon Web Service**, ofrece una serie de servicios para que los desarrolladores puedan manejar máquinas virtuales en la nube, las cuales también hacen las veces de espacio de almacenamiento.

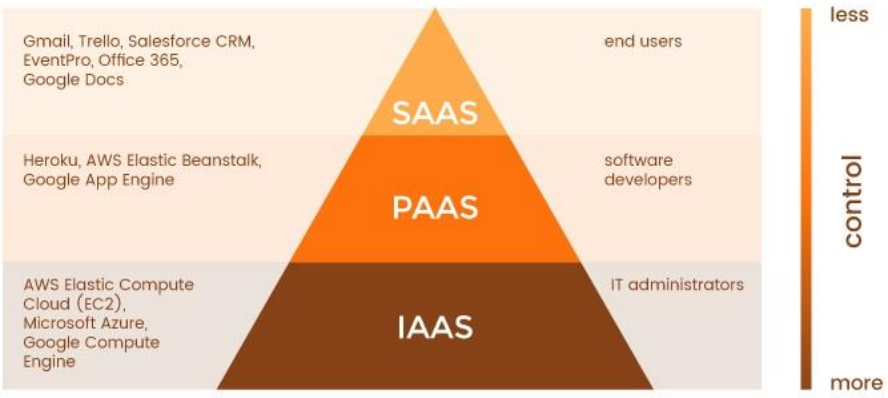
PaaS (Platform-as-a-Service) alternativa idónea para desarrolladores de aplicaciones que quieren preocuparse de construir la app. La infraestructura la **proporciona la plataforma y se ocupa tanto de su gestión como de su mantenimiento**.

Se encarga de **mantener automática la escalabilidad**, haciendo uso de un mayor número de recursos en caso de que sea necesario. Los desarrolladores tienen que intentar que sus aplicaciones estén lo mejor optimizadas posible para no consumir demasiados recursos.

Un **ejemplo** es **Jelastic**, una plataforma que puede crear aplicaciones con soporte de Java, PHP, Node.js, Ruby, Python, Docker y Kubernetes.

SaaS (Software-as-a-Service) servicio basado en la web, por **ejemplo**, el **Webmail de Gmail**. **Los usuarios acceden al servicio sin prestar atención al software**. El desarrollo, mantenimiento y resto de gestiones son responsabilidad del proveedor.

[3]Así, los usuarios tienen un control mínimo sobre el servicio en cuestión. Ellos se sitúan en la capa más superficial del mismo.



[3] Esquema de comparativa de Saas, PaaS e IaaS.

Semejanzas:

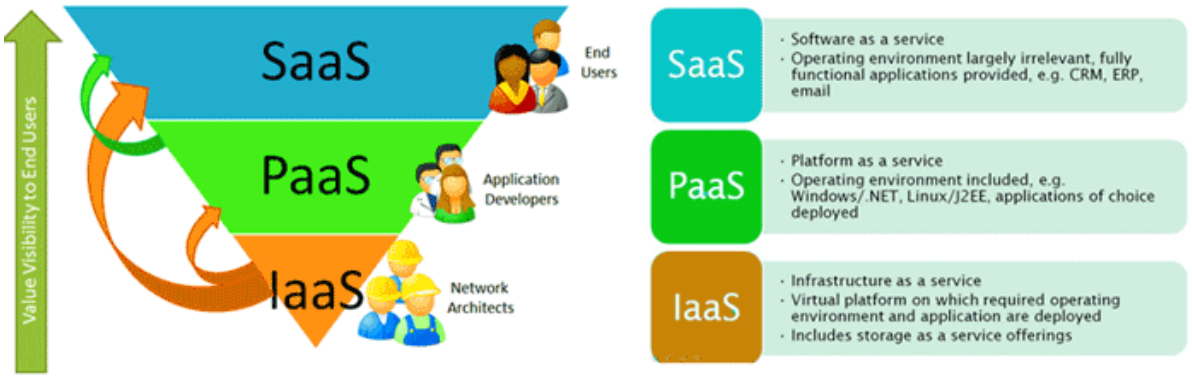
Todos ellos son servicios en línea, cuyo funcionamiento se da a través de la **nube**.

Todos ofrecen la posibilidad de pago por recursos y uso.

Todos ellos son sistemas escalables, de modo que permiten aumentar la capacidad.

Diferencias:

[4]Tiene que ver con el mantenimiento y soporte ofrecidos por el proveedor. En IaaS es el desarrollador debe ocuparse de todo. En SaaS, los usuarios ni siquiera tienen acceso al software. En PaaS pueden gestionar la plataforma, pero no el servidor.



[4] Esquema de comparativa de Saas, PaaS e IaaS.

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

Apache Tomcat **funciona como un contenedor de servlets**. I**mplementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation**. Fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Especificaciones de servlet	Especificaciones JSP	EL Spec	Especificaciones de WebSocket	Especificación de autenticación (JASIC)	Versión de Apache Tomcat	Última versión publicada	Versiones de Java compatibles
5.0	3.0	4.0	2.0	2.0	10.0.x	10.0.0-M8 (alfa)	8 y posteriores
4.0	2.3	3.0	1.1	1.1	9.0.x	9.0.38	8 y posteriores
3.1	2.3	3.0	1.1	1.1	8.5.x	8.5.58	7 y posteriores
3.1	2.3	3.0	1.1	N / A	8.0.x (reemplazado)	8.0.53 (reemplazado)	7 y posteriores
3.0	2.2	2.2	1.1	N / A	7.0.x	7.0.106	6 y posterior (7 y posterior para WebSocket)
2.5	2.1	2.1	N / A	N / A	6.0.x (archivado)	6.0.53 (archivado)	5 y posteriores
2.4	2.0	N / A	N / A	N / A	5.5.x (archivado)	5.5.36 (archivado)	1.4 y posterior
2.3	1.2	N / A	N / A	N / A	4.1.x (archivado)	4.1.40 (archivado)	1.3 y posterior
2.2	1.1	N / A	N / A	N / A	3.3.x (archivado)	3.3.2 (archivado)	1.1 y posterior

[5]

[5] Tabla de versiones de Apache Tomcat

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

URI: (Uniform Resource Identifier -Identificador Uniforme de Recursos) es una **cadena de caracteres que identifica los recursos de una red** de forma unívoca. **Normalmente estos recursos son accesibles en una red o sistema**. Las URI engloban los otros dos conceptos, pudiendo clasificarlas entre URL, URN o ambas.

Un URI consta de las siguientes **partes**:

Esquema: nombre para asignar los identificadores*, e.g. urn:, tag:, cid:.* También identifica el protocolo de acceso al recurso, *http:, mailto:, ftp:, etc.*

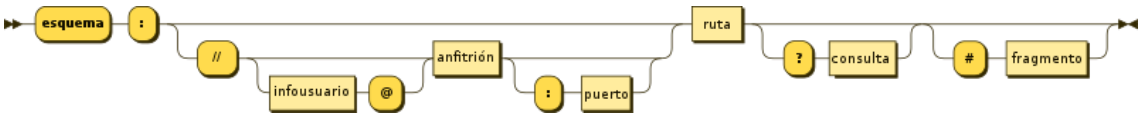
Autoridad: elemento jerárquico que identifica la autoridad de nombres, *//www.example.com*.

Ruta: información usualmente organizada en forma jerárquica, que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres, *e.g. /domains/example*.

Consulta: información con estructura no jerárquica que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres. Comienza mediante el carácter '?'.

Fragmento: Permite identificar una parte del recurso principal, o vista de una representación de este. El comienzo se indica mediante el carácter '#'.

Un URI **se diferencia de un URL en que permite incluir en la dirección una subdirección, determinada por el "fragmento".**



6

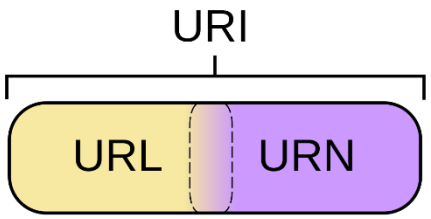
URL: (Uniform Resource Locator -Localizador Uniforme de Recursos) Son unas **cadenas de texto que se usan para nombrar recursos en Internet para su localización**, **cuyos recursos referidos pueden cambiar**, esto es, la dirección puede apuntar a recursos variables en el tiempo.

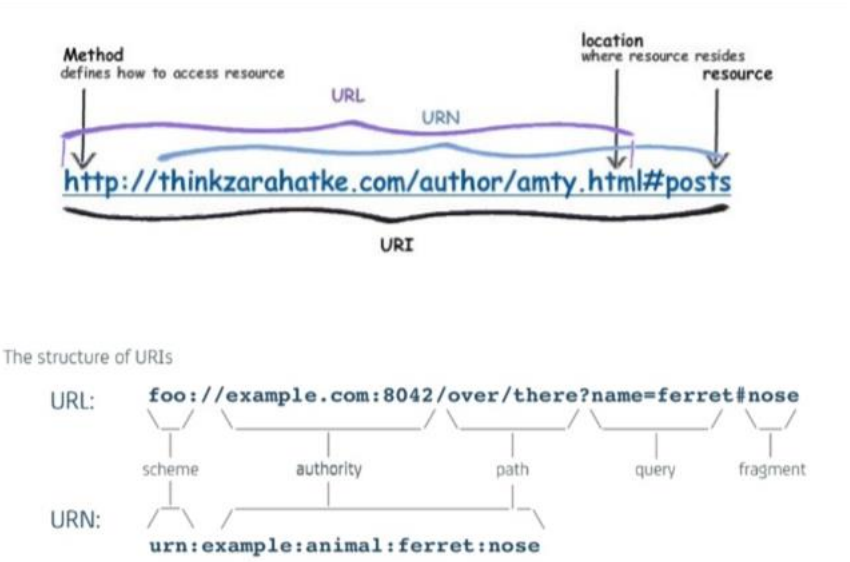
Ejemplo: http://es.wikipedia.org:80/wiki/Special:Search?search=tren&go=Go

[6] Esquema de las partes que compone un Identificador Uniforme de Recurso (URI).

URN: (Uniform Resource Name -Nombre Uniforme de Recursos) Son unas **cadenas de texto que se usan para nombrar recursos en Internet para su identificación**, **no indican exactamente dónde se encuentra ese objeto**.

Ejemplo: urn:isbn:0451450523



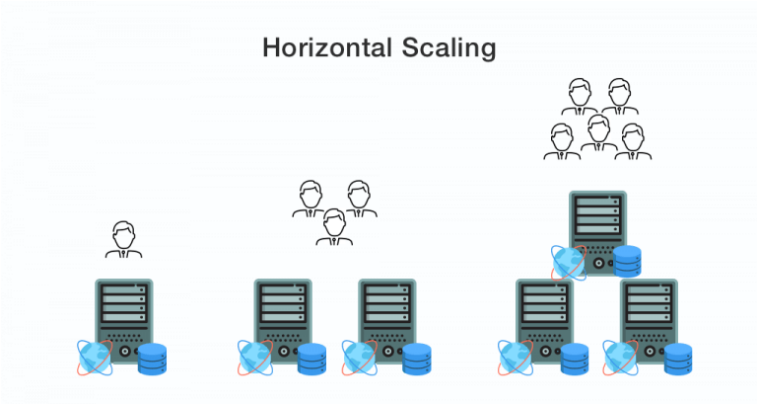


7

[7] Esquemas comparativos entre URL y URN.

9. Escalabilidad horizontal: Ventajas e inconvenientes.

[8]El escalamiento horizontal es sin duda el **más potente**, pero también el **más complicado**. Este modelo **implica tener varios servidores** (conocidos como **Nodos**) **trabajando como un todo**. **Se crea una red** de servidores conocida como **Cluster**, **con la finalidad de repartirse el trabajo entre todos**, cuando su performance se ve afectada con el **incremento de usuarios**, **se añaden nuevos nodos al cluster**, de esta forma **a medida que son requeridos**, más y más nodos **son agregados**.



Para que el escalamiento horizontal **funcione deberá existir un servidor primario desde el cual se administra el cluster**. Cada servidor deberá tener un software que le permite integrarse.

Ventajas:

- El crecimiento casi infinito, incorpora cuantos servidores sean necesarios.
- Es posible combinarse con el escalamiento vertical.
- Soporta la alta disponibilidad.
- Si un nodo falla, los demás siguen trabajando.
- Soporta el balanceo de cargas.

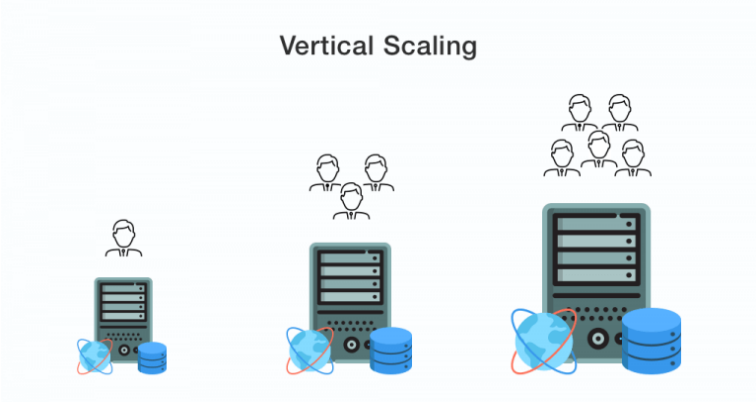
Desventajas:

- Requiere de mucho mantenimiento.
- Es difícil de configurar.
- Grandes cambios en las aplicaciones (al no tener diseño para los cluster).
- Requiere de una infraestructura más grande.

[8] Esquema para aumento de nodos

10. Escalabilidad vertical: Ventajas e inconvenientes.

[9]La escalabilidad vertical es **más simple**, pues significa **aumentar el hardware de uno de los nodos por uno más potente**, como disco duro, memoria, procesador. **También puede ser la migración completa del hardware** por uno más potente. **El esfuerzo** de este crecimiento **es mínimo**, pues no tiene repercusiones en el software, ya que **solo será respaldar y migrar los sistemas al nuevo hardware**.



Tiene algunos **aspectos negativos**, ya que nuestro **crecimiento está ligado al hardware**, y este, tarde o temprano tendrá un límite. Ahora bien, no significa que este modelo de escalamiento sea malo, ya que **lo podemos combinar con el escalamiento horizontal para obtener mejores resultados.**

Ventajas:

- No implica un gran problema para las aplicaciones, pues todo el cambio es sobre el hardware.
- Es mucho más fácil de implementar que el escalamiento horizontal.
- Puede ser una solución rápida y económica.

Desventajas:

- El crecimiento está limitado por el hardware.
- Una falla en el servidor implica que la aplicación se detenga.
- No soporta la Alta disponibilidad.
- Hacer un upgrade del hardware al máximo puede llegar a ser muy caro.

[9] Esquema para aumento de hardware

CONCLUSIÓN

Con la realización de este estudio mediante preguntas, hemos conseguido tener un estudio más profundo al contrastar información de las diferentes páginas consultadas en la realización de las cuestiones a tratar.

BIBLIOGRAFIA

Pregunta 2

https://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html

Pregunta 3

https://einatec.com/tipos-de-aplicaciones-web/

Pregunta 4

https://instintobinario.com/arquitectura-en-tres-capas/

Pregunta 5

https://blog.infranetworking.com/servidor-litespeed/

Pregunta 6

https://axarnet.es/blog/saas-paas-iaas#:~:text=La%20principal%20diferencia%20entre%20SaaS,siquiera%20tienen%20acceso%20al%20software.

Pregunta 7

http://tomcat.apache.org/whichversion.html

Pregunta 8

https://es.wikipedia.org/wiki/Identificador_de_recursos_uniforme

Preguntas 9 y 10

https://www.grapheverywhere.com/escalabilidad-de-bases-de-datos/