

AutoExamen: Servicios

1. HTTP. Diferencias con HTTPS
 2. FTP
 3. Qué es un clúster y sus ventajas
 4. Acoplamiento
 5. Clasificación según servicios y tipos configuración
-

Ejercicio 1. HTTP es un protocolo de la capa de aplicación orientado a las transacciones utilizando el esquema de petición-respuesta entre un cliente y un servidor, siendo su uso más común, el de hacer llegar a los usuarios un documento html cuando los usuarios solicitan-por lo general- dicho documento alojado en un nombre de dominio que utiliza las nomenclaturas DNS que identifican a la dirección IP de la máquina que contiene el documento html.

Dentro del protocolo HTTP, podemos identificar a los clientes, que son denominados como user agent y la información transmitida se le denomina recurso. Un recurso puede ser desde una consulta a una base de datos o el resultado de la ejecución de un programa hasta un archivo. Los recursos se identifican por una URL.

HTTP es un protocolo sin estado, es decir, que no guarda información sobre conexiones anteriores. No obstante, actualmente las páginas webs requieren de la información de conexiones anteriores, por lo que se introduce el concepto de cookies y con éstas se implementa la noción de sesiones, para poder almacenar información de conexiones anteriores.

Como cualquier protocolo, HTTP dispone de sus respectivas versiones. Las versiones del protocolo HTTP son las siguientes:

- HTTP/1.0. Es la versión más utilizada. Sus conexiones son no persistentes.
- HTTP/1.1. Versión más actualizada y mejorada respecto a HTTP/1.0. Sus conexiones son persistentes.

Otro punto importante a mencionar sobre HTTP, son sus etapas de comunicación. Sus etapas de comunicación son 3:

1. El usuario escribe en la barra de direcciones del navegador el nombre del recurso al que quiere acceder
2. El navegador web divide en tres partes la URL que identifica al recurso solicitado
 - a. Protocolo. En este caso "http".
 - b. Nombre del servidor. "Examenservicios.com", por ejemplo
 - c. Camino. "/pregunta1examen"
3. El navegador contacta con el servicio de nombres para resolver el nombre del recurso por la dirección IP que identifica a la máquina servidora.

No obstante, estas son las etapas para la conexión. Cuando un cliente llega a la web y encuentra, por ejemplo en <http://www.examenservicios.com/pregunta1examen> 3 imágenes, se suceden los siguientes pasos.

1. El cliente establece la conexión TCP con el servidor <http://www.examenservicios.com/> por el puerto 80.
2. El servidor que está a la escucha por el puerto 80, acepta la conexión y se lo reporta al cliente.
3. El cliente solicita al servidor el documento fasescomunicacionhttp.html dentro de <http://www.examenservicios.com/pregunta1examen/>
4. El servidor creará la respuesta con el html solicitado y se lo enviará al usuario
5. El servidor cerrará la conexión TCP.
6. El cliente se da cuenta que en fasescomunicación.html hay 3 imágenes una vez que lo recibe.
7. Se repiten los pasos del 1-6 por cada imagen.

Como se puede observar, a cada petición, le sucede una respuesta, por lo que se tendrá que analizar y mencionar la estructura de la petición o solicitud y la respuesta.

La solicitud se compone de:

- Líneas de solicitud. Que contine
 - ◆ Método. POST por ejemplo
 - ◆ Dirección URL o Host. www.academialopedevega.org
 - ◆ Versión del protocolo. HTTP/1.1
- Campos del encabezado de la solicitud. Son líneas opcionales que aportan información extra sobre la petición y/o cliente. If-modified-since, por ejemplo.
- Cuerpo de la solicitud. Envío de datos por un comando POST durante la transmisión

Por otro la respuesta se compone de:

- Líneas de estado. Que contiene
 - ◆ Versión del protocolo. HTTP/1.1 por ejemplo
 - ◆ Código estado. 200 por ejemplo
 - ◆ Significado del código de estado. Ok
- Campos del encabezado de respuesta. Líneas opcionales que aportan información extra sobre la respuesta y/o servidor.
- Cuerpo de respuesta. Contiene el documento solicitado.

Para concluir con HTTP, se puede realizar una rápida mención de los códigos de estado, siendo

- 10x. Mensajes informativos
- 20x. Éxito
- 30x. Redirección
- 40x. Error en el cliente
- 50x. Error en el servidor.

Pero esto es HTTP que por su característica de enviar los datos en texto plano, se convierte en un protocolo de transacción inseguro, ya que cualquier atacante podría capturar tráfico y apropiarse de los datos transferidos como credenciales. Para solucionar esto surgió el protocolo HTTPS

HTTPS es un protocolo seguro de transacciones basado en HTTP. HTTPS se caracteriza por enviar los datos cifrados con SSL/TLS en tránsito, lo que lo convierte en un protocolo de transacciones de información sensible más apropiado que HTTP.

Sus diferencias con HTTP son las siguientes.

	HTTP	HTTPS
URL	Comienzan por http://	Comienza por https://
PUERTO	Puerto 80 u 8080	Puerto 443
SEGURIDAD	Protocolo inseguro ya que envía la información en texto plano, por lo que cualquier atacante realizando un esquema de MiTM podría apropiarse de los datos transferidos	Protocolo que surgió para enviar información sensible de forma segura en tránsito

Para implementar HTTPS, se requiere de un certificado digital que deberá estar firmado por una entidad certificadora, para garantizar que el servidor es realmente quien dice ser y evitar así posibles ataques de Phishing.

NOTA: 2 PUNTOS

Ejercicio 2. FTP es un protocolo de la capa de transporte destinado a la transferencia de archivos entre equipos conectados a una red TCP/IP y con una arquitectura cliente-servidor.

[Ofrece la máxima velocidad pero no la máxima seguridad, ya que permite que un posible atacante acceda y se apropie de los datos transferidos] -0.5

La forma de comunicación de FTP se basa en que el Intérprete del Protocolo [PI] del usuario establece la conexión por un control de conexión en el puerto 21. Las órdenes estándar son generadas por el usuario que se las envía al proceso servidor por el control de conexión. Estas órdenes definen los parámetros y naturaleza de la conexión. Por otro lado, las respuestas estándar son enviadas desde el PI servidor al PI usuario por el control de conexión como respuesta a las órdenes.

Un proceso de datos DTP u otro proceso, deberá esperar a que el servidor establezca la conexión por el puerto especificado y en función de los parámetros especificados anteriormente.

Todo este proceso es independiente al sistema de archivos utilizado, por lo que no importa el sistema operativo de cada ordenador, además estas conexiones son bidireccionales.

Los modos de conexión de FTP son 2

- Modo activo. El servidor estará a la escucha en el puerto 20 y el cliente en un puerto con un número aleatorio mayor de 1024. El cliente enviará un comando PORT al servidor para que pueda establecer la conexión. Este modo de conexión es inseguro, ya que obliga al cliente a estar escuchando en puertos mayores a 1024, y esto en una red insegura como Wifi, podría permitir al atacante acceder, capturar tráfico y apropiarse de datos transferidos. Para solucionar esto surgió el modo pasivo
- Modo pasivo. El cliente envía un comando PASV al servidor y este le enviará un puerto con número mayor a 1023. El cliente usará el siguiente puerto al enviado.

NOTA: 1.5 PUNTOS

Ejercicio 3. Un clúster es un conjunto de máquinas unidas por una red de comunicación y trabajando por un servicio conjunto. Los clústers presentan las siguientes ventajas:

1. Disponibilidad. Tienen la capacidad de estar operativos 24x7x365 días a un cuando algún ordenador o nodo del clúster no lo esté.
2. Distribución en paralelo.
 - a. Flexibilidad. Los balanceadores de carga no están sujetos a ninguna arquitectura en concreto a nivel hardware
3. Escalabilidad. Capacidad de hacer frente a volúmenes de trabajo cada vez mayores a nivel óptimo de rendimiento
4. Expansibilidad. Capacidad de aumentar las capacidades del clúster.
5. Transferencia de información o cualquier servicio a una alta velocidad, bajo en costes y de forma ininterrumpida.
6. Incremento de velocidad de procesamiento ofrecido por clústers de alto rendimiento.
7. Incremento del número de transacciones ofrecido por clústers de balanceo de carga.
8. Incremento de confiabilidad ofrecido por clústers de alta disponibilidad.
9. Costos. Implementación y diseño presenta unos costes más bajos en comparación con las otras alternativas.

NOTA: 2 PUNTOS

Ejercicio 4. Acoplamiento. El acoplamiento es la integración de los elementos de software de cada nodo del clúster. El acoplamiento se ve favorecido por los procesos de comunicación de sus nodos, por lo que se define el acoplamiento. El acoplamiento puede ser de 3 tipos.

1. Fuerte. Tipo de acoplamiento en el que los nodos del clústers están muy relacionados. Un ejemplo sería un Windows Server con Office y un Windows 7 con Office, ambos presentan un acoplamiento fuerte. *[Posibilitan mayoría funcionalidades cluster de manera altamente cooperativa] -1punto* Es importante tener acceso a todas las partes del sistema operativo estrechamente relacionadas y distribuido entre los nodos.
2. Medio. No es necesario un conocimiento tan exhaustivo todos los recursos de todos los nodos aunque siguen usando software de otros nodos para aplicaciones de bajo nivel. No requiere de sistema de nombres común entre nodos
3. Débil. Son aplicaciones construidas por librerías o bibliotecas preparadas para aplicaciones distribuidas.

NOTA: 1 PUNTO

Ejercicio 5. La clasificación de los clústers según sus servicios puede ser:

- Clústers de alto rendimiento. Son aquellos que ejecutan tareas que necesitan una gran capacidad computacional o una gran cantidad de memoria o ambos a la vez. *[Estas tareas pueden comprometer recursos por largos periodos de tiempo] -0.5 puntos*
- **Alta eficiencia.** Clústers cuyo objetivo de diseño es el de ejecutar el mayor número de servicios en el menor tiempo posible. En este tipo de clústers, el posible retardo entre nodos no se considera un problema.
- **Alta disponibilidad.** Su objetivo es el de estar disponible 24x7x365 días al año. Tratará de lograr la alta disponibilidad y confiabilidad de los recursos y servicios ofrecidos. Esto lo puede lograr con software de detección de errores y software que permita reparar los errores que puedan ocurrir.
- **Balanceo de carga.** Conjunto de servidores que se reparten la carga de trabajo formando
 - Front-end. Uno o más nodos que se reparten peticiones que recibe el cluster a otros ordenadores
 - Back-end. Esos otros ordenadores a los que se les envía las peticiones que recibe el clúster.

Los tipos de configuración dentro de los clústers de alta disponibilidad pueden ser de 2 tipos.

- **Configuración Activo/activo.** Todos los servidores del clústers tienen acceso a los mismos recursos y podrán ejecutarlos independientemente de los demás clústers simultáneamente. Este tipo de configuración trae como ventaja que en caso de caída de alguno de los nodos. El servicio seguirá disponible ya que los demás nodos tienen acceso a todos los recursos y podrán seguir ofreciéndoles pues trabajan todos a la vez, aunque cuando cae un nodo, la carga de trabajo del nodo caído recae sobre los demás nodos, provocando cierta degradación de los recursos hasta que el nodo se recupere y vuelva a la normalidad.
- **Configuración activo/pasivo.** Servidor activo tiene acceso a los recursos y los demás [pasivos] también tienen acceso a los recursos aunque no los activan hasta que el propietario deje de estar disponible. Como ventaja presenta que no se degrada y solamente se reinicia cuando el servidor activo no está disponible, aunque trae consigo importantes desventajas como que los servidores pasivos no ofrecen recursos mientras están a la espera o que se tarda un tiempo relativamente alto en migrar los recursos al nodo en espera..

NOTA:1.5 PUNTOS

EXTRA:

Analizando Petición-respuesta HTTP

```
GET / HTTP/1.1
Host: academialopedevega.org
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8,ro;q=0.7
Cookie: MOODLEID1_=%2591%25BC%25CBI%259DV%25CF%25B3
```

Esta es la petición que se ha realizado por HTTP. Si la analizamos con la teoría, veremos lo siguiente.

Líneas de petición



En este caso se analizará el método GET ya que es el usado. El método GET solicita el recurso ubicado en la URL especificada, en este caso especifica el recurso ubicado en academialopedevega.org.

Campos de encabezado de petición

```
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Safari/537.36
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8,ro;q=0.7
Cookie: MOODLEID1_=%2591%25BC%25CBI%259DV%25CF%25B3
```

Aquí se puede ver que, tal y como se ha visto en la teoría, aporta información extra sobre la petición y/o cliente. Los ejemplos más claros está en el parámetro *Connection*, que se encuentra *keep-alive* o el User-agent [antes visto que es el usuario, el cliente].

Analizando la respuesta.

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 02 Dec 2017 11:37:38 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.6.32
Vary: Cookie,Accept-Encoding
Composed-By: SPIP 3.0.20 @ www.spip.net +
spip(3.0.20),compagnon(1.4.1),dump(1.6.9),images(1.1.10),forum(1.8.40),jqueryui(1.8.21),mediabox(0.8.11),mots(2.4.13),ms
ie_compat(1.2.0),organiseur(0.8.12),petitions(1.4.6),porte_plume(1.12.4),revisions(1.7.9),safehtml(1.4.1),sites(1.7.13),
squelettes_par_rubrique(1.1.1),stats(0.4.28),svp(0.80.26),tw(0.8.30),urls(1.4.26),vertèbres(1.2.2),calendriermini(2.3.8)
,agenda(3.14.10),escal(3.80.1),cookiebar(1.1.4),seo(2.1.5),adaptive_images(1.9.1),forms(0.5.6),iterateurs(0.6.1),queue(0
.6.6),breves(1.3.6),compresseur(1.8.11),medias(2.7.66)
X-Spip-Cache: 86400
Last-Modified: Sat, 02 Dec 2017 11:37:38 GMT
X-Powered-By: PleskLin
X-Powered-By: PleskLin

1ca8
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="es" >

  <head><title>CES Lope de Vega SCA</title>
  <meta name="Keywords" content="CES,ense..anza,Lope de Vega,Cordoba,ciclos formativos,bachillerato" />
  <meta name="author" content="CES Lope de Vega SCA" />
  <!--[if !IE]><!--><style type="text/css">img.adapt-img,.lazy img.adapt-img{opacity:0.70;max-width:
  100%;height:auto;}.adapt-img-wrapper,.adapt-img-wrapper:after{display:inline-block;max-width:100%;position:relative;-
  webkit-background-size:100% auto;background-size:100% auto;background-repeat:no-repeat;line-height:1px;}html body
  .adapt-img-wrapper.lazy,html.lazy body .adapt-img-wrapper,html body .adapt-img-wrapper.lazy:after,html.lazy body .adapt-
```

La respuesta a la petición de conexión se compone por lo mostrado en la captura anterior. Si se analiza cada parte más en concreto, encontramos:

Líneas de estado



Campos de encabezado de respuesta


```

Server: nginx
Date: Sat, 02 Dec 2017 11:37:38 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.6.32
Vary: Cookie,Accept-Encoding
Composed-By: SPIP 3.0.20 @ www.spip.net +
spip(3.0.20),compagnon(1.4.1),dump(1.6.9),images(1.1.10),forum(1.8.40),jqueryui(1.8.21),mediabox(0.8.11),mots(2.4.13),ms
ie_compat(1.2.0),organiseur(0.8.12),petitions(1.4.6),porte_plume(1.12.4),revisions(1.7.9),safehtml(1.4.1),sites(1.7.13),
squelettes_par_rubrique(1.1.1),stats(0.4.28),svp(0.80.26),tw(0.8.30),urls(1.4.26),vertbres(1.2.2),calendriermini(2.3.8)
,agenda(3.14.10),escal(3.80.1),cookiebar(1.1.4),seo(2.1.5),adaptive_images(1.9.1),forms(0.5.6),iterateurs(0.6.1),queue(0
.6.6),breves(1.3.6),compresseur(1.8.11),medias(2.7.66)
X-Spip-Cache: 86400
Last-Modified: Sat, 02 Dec 2017 11:37:38 GMT
X-Powered-By: PleskLin
X-Powered-By: PleskLin

```

En la teoría se dice que aportan información extra sobre respuesta y/o servidor, y encontramos parámetros como el *Server* o la fecha.

Cuerpo de respuesta

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="es" >

  <head><title>CES Lope de Vega SCA</title>
  <meta name="Keywords" content="CES,ense..anza,Lope de Vega,Cordoba,ciclos formativos,bachillerato" />
  <meta name="author" content="CES Lope de Vega SCA" />
  <!--[if !IE]><!--><style type="text/css">img.adapt-img,.lazy img.adapt-img{opacity:0.70;max-width:
  100%;height:auto;}.adapt-img-wrapper,.adapt-img-wrapper:after{display:inline-block;max-width:100%;position:relative;-
  webkit-background-size:100% auto;background-size:100% auto;background-repeat:no-repeat;line-height:1px;}html body
  .adapt-img-wrapper.lazy,html.lazy body .adapt-img-wrapper,html body .adapt-img-wrapper.lazy:after,html.lazy body .adapt-
  img-wrapper:after{background-image:none}.adapt-img-wrapper:after{position:absolute;top:0;left:0;right:0;bottom:
  0;content:""}@media print{html .adapt-img-wrapper{background:none}html .adapt-img-wrapper img {opacity:1}html .adapt-
  img-wrapper:after{display:none}}</style>

```

Y finalmente en el cuerpo encontramos el documento html.