MEDAC
Instituto Oficial de Formación Profesional

DESPLIEGUE DE APLICACIONES WEB
TÉCNICO EN DESARROLLO DE APLICACIONES WEB

Introducción a los servidores web

02

ÍNDICE

OBJETIVOS

Conocer el funcionamiento de un servidor web

Conocer los servidores web más utilizados

Ser capaz de instalar y configurar un servidor web

Ser capaz de instalar y configurar PHP y MySQL

Tener un primer contacto con el sistema Docker

/ 1. Introducción y contextualización práctica

En este tema, aprenderemos los fundamentos y protocolos necesarios para entender cómo trabaja un servidor web y posteriormente ser capaces de instalar y configurar servicios básicos.

Hablaremos de Docker, un proyecto que nos servirá para automatizar el despliegue de aplicaciones de forma rápida y optimizando el uso de recursos.

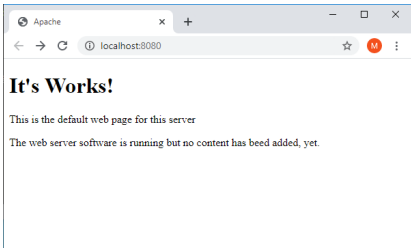

Fig. 1. Servidor web Apache sin configurar

Planteamiento del caso práctico inicial

Escucha el siguiente audio donde planteamos la contextualización práctica de este tema; encontrarás su resolución en el apartado “Resumen y Resolución del caso práctico”.

Audio Intro. “Modernización de una web estática”
https://bit.ly/38WzuRn

/ 2. Funcionamiento de un servidor web

Un servidor web puede definirse como un software preparado para escuchar permanentemente peticiones de los clientes y ofrecer la respuesta correcta; ya sea entregando la página web al navegador o listando un mensaje de error si la petición es incorrecta. Se llama también servidor web al equipo físico que realiza estas funciones.

Como vimos en el tema 1, la finalidad de un servidor web es el intercambio de archivos HTML, pudiendo necesitar software complementario que le permita gestionar ciertas peticiones complejas.

Fig. 2. Logo PHP

Para solucionar esta posible carencia, instalamos otros componentes para conseguir que el servidor web pueda procesar código escrito en PHP que, a diferencia de HTML, no se ejecuta en el cliente, si no en el servidor.

En el año 1994, Berners-Lee fundó el W3C (**World Wide Web Consortium**), una organización cuyo objetivo fundamental es **regular las recomendaciones y los estándares utilizados, asegurando la compatibilidad y la evolución de los servicios web.**

Cuando hablamos de ejecución en el cliente nos referimos a aquel código que se puede abrir directamente en un navegador web, por ejemplo: HTML, CSS o un programa escrito en JavaScript.

Por el contrario, existen otros programas que requerirán un preprocesador en el servidor que haga posible que el código se ejecute, y así resolver correctamente la petición del cliente. Entre los lenguajes de programación que se ejecutan en el lado del servidor podemos encontrar PHP, Python, Perl o Ruby.

Podemos encontrar más información acerca de los lenguajes que se ejecutan en el servidor en el siguiente enlace:

https://www.hostinet.com/formacion/general/lenguajes-del-lado-servidor-o-cliente/

De la misma manera que necesitamos un software adicional para procesar PHP, necesitaremos un programa que gestione la base de datos y nos garantice la integridad de los datos que tratamos. De esta tarea se encarga MySQL, uno de los gestores de bases de datos relacionales más utilizados.

Como alternativa a MySQL tenemos MariaDB, PostgreSQL o SQLite.

Podemos leer más acerca de estos gestores de bases de datos en el siguiente enlace:

https://www.inesem.es/revistadigital/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/

Fig. 3. Logo MySQL

/ 3. Características de los servidores web libres y propietarios

En este apartado hablaremos de las características de los que posiblemente sean los servidores web más utilizados: **Apache, Nginx y Microsoft IIS**.

Antes de empezar a analizar cada uno de estos servidores debemos valorar un aspecto muy importante en cualquier software: el tipo de licencia.

Según el tipo de licencia que utilicen encontraremos dos tipos:

- Servidores web libres o de **código abierto** (Open Source): permiten su uso de manera gratuita, como es el caso de **Apache y Nginx.**
- Servidores web propietarios: requieren una licencia para poder usarlo, como por ejemplo Microsoft IIS o Nginx Plus, la versión comercial de Nginx.

Fig. 4. Open Source logo

3.1. Apache

Apache es un servidor web Open Source multiplataforma. Desde que apareció en abril del año 1995, Apache se ha convertido en el servidor web más utilizado en todo el mundo.

En su web oficial, Apache se define a sí mismo con los siguientes términos:

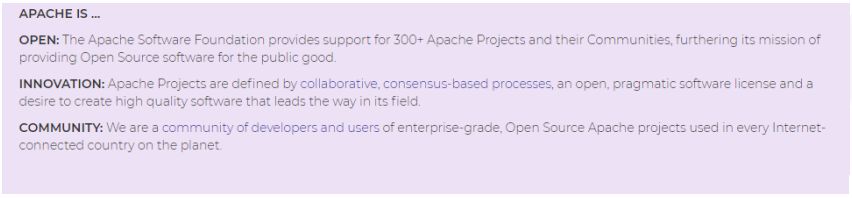

Fig. 5. Extracto de Apache.org

3.2. Nginx

Nginx es un servidor web también con licencia Open Source, que destaca por su alto rendimiento. Al igual que Apache es multiplataforma y se puede instalar en Windows, Linux y Unix.

La primera versión se liberó en octubre de 2004, y desde entonces su uso ha crecido exponencialmente. En el apartado "casos de éxito" de la web oficial de Nginx, podemos ver qué grandes empresas confían en la versión libre para levantar sus servicios web: https://www.nginx.com/success-stories/

3.3. Microsoft IIS

Cuando hablamos de IIS (Internet Information Services) nos referimos al servidor web de Microsoft; cuya primera versión apareció con Windows NT 3.51, en mayo del año 1995.

Su instalación se realiza sobre un S.O. con licencia comercial, por lo que IIS es un servidor web **propietario** (aunque a diferencia de Nginx Plus, no conlleva ningún coste adquirir el producto para su uso).

Internet Information Services se instala como "**característica**" desde el "panel de control" en "sistemas de escritorio" o como "**rol**" en las versiones Windows Server.

Fig. 6. Index por defecto de IIS

3.4. Comparativa de servidores web

Posible

Para comprender las características de estos servidores web en su conjunto, mostramos una tabla comparativa en la que se ilustran los servidores web más importantes de la actualidad.

	Apache	Nginx	IIS
Open Source	SI	SI	NO
SSL/TSL	SI	SI	SI
Proxy Inverso	SI	SI	SI
Primera aparición	1995	2004	1995
PHP Nativo	NO	NO	NO
Multiplataforma	SI	SI	NO

Tabla 1. Comparativa de los servidores web más importantes

Audio 1. "Servidores web"
https://bit.ly/2Otub2q

/ 4. Caso práctico 1: "Instalación Apache en Windows"

Planteamiento: Miguel es el responsable de sistemas de una empresa de transportes. Acaban de desarrollar una web y deciden alojarla en un equipo bajo Windows.

Nudo: ¿Sería posible? ¿Cómo podría implementarlo?

Desenlace: Como hemos visto en el punto 3.1, Apache es multiplataforma y, aunque habitualmente se implemente sobre sistemas Linux, también podríamos disfrutar de la fiabilidad de este servidor web en Windows.

Para agilizar la instalación de Apache, MySQL y PHP utilizaremos WAMP, que es un paquete que incluye los tres componentes:

https://wampserver.com/en/

(Puedes apoyarte en el vídeo 1 de este tema, para instalar y configurar Wamp Server en Windows).

Una vez completada la instalación, comprobaremos que Apache está funcionando y veremos el index.html de WAMP.

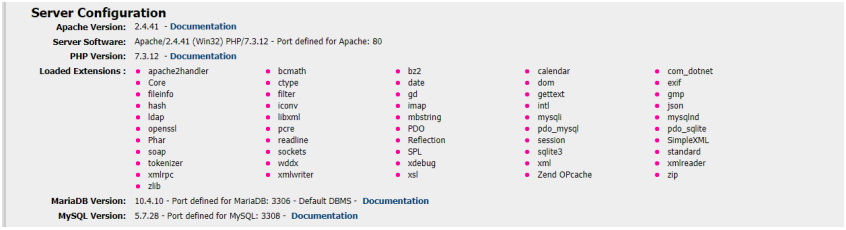

Fig. 7. Index.html de Wamp Server

A continuación, comprobaremos que PHP estaría funcionando correctamente. Para ello, accederemos a la carpeta raíz del servidor **(C:\wamp\www)** y crearemos un archivo llamado **info.php** con el siguiente contenido:

```
<?php

phpinfo();

phpinfo(INFO_MODULES);

?>
```

Fig. 8. info.php

Al crear esto en nuestro servidor web, podremos acceder mediante un navegador a la siguiente URL:

http://localhost/info.php.

Si creamos el archivo en una subcarpeta, la ruta de acceso será diferente.

Si todo ha ido bien, nos aparecerá una página informativa indicando la versión de PHP que se está utilizando, los módulos de los que dispone, así como diversa información relacionada con las características de PHP.

/ 5. Instalación de servidores web

En los siguientes apartados veremos cómo realizar la instalación básica de Apache, Nginx y IIS.

Tanto Apache y Nginx son multiplataforma, y aunque se pueden instalar en Windows, vamos a centrarnos en la instalación en sistemas operativos Linux. Este tándem nos permitirá tener un servidor web a un coste relativamente bajo, evitando así el sobrecosto de las licencias.

Lo más normal es utilizar servidores virtuales en Clouds, aunque podríamos valorar soluciones más asequibles como utilizar ordenadores de bajo coste, como podría ser una Raspberry Pi para montar un servidor web. Debido a su bajo consumo eléctrico y buen rendimiento, la Raspberry Pi podría ser una opción bastante decente en entornos de desarrollo o aprendizaje, pero debemos tener en cuenta las limitaciones de la arquitectura del procesador (ARM)

En el siguiente enlace encontrarás más información sobre qué es una Raspberry Pi, y cómo montar un servidor web Apache sobre ella. https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md

Vídeo 1. "Instalación LAMP Server"
https://bit.ly/3exolHC

5.1. Instalación de Microsoft IIS en Windows 10

A diferencia de Apache o Nginx, IIS no es multiplataforma y para poder instalarlo necesitaremos un sistema operativo Windows.

Dependiendo de si lo instalamos en una versión de escritorio o en la modalidad servidor, el proceso variará; pero en cualquier caso la carpeta raíz del servidor se establecerá en "**C:\inetpub\wwwroot**" y podremos gestionar las opciones desde el **Administrador de Internet Information Services** (IIS)

Instalar IIS en Windows 10 es tan fácil como ir a Panel de Control > Programas > **Activar o desactivar características de Windows y activarlo.**

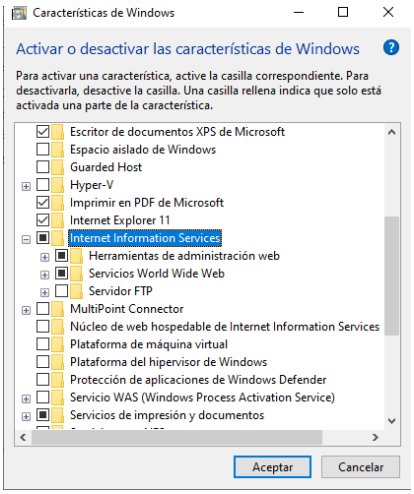

Fig. 9. Instalación IIS Windows 10

5.2. IIS en Windows Server

En sistemas Windows Server, la instalación la haremos desde el administrador del servidor y pulsaremos el botón "**agregar roles y características**". Seleccionaremos una instalación basada en características o en roles.

Nos aparecerá un menú con todos los roles disponibles, y activaremos **Servidor web** (IIS). En las siguientes ventanas nos permitirá añadir funcionalidades al servidor, pero de momento con la instalación por defecto será suficiente.

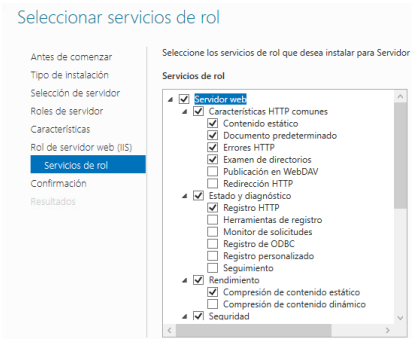

Fig. 10. Instalación IIS

El siguiente paso sería instalar **MySQL y PHP.** Para ello utilizaremos una herramienta de Microsoft llamada **Microsoft Web Platform Installer.** En esta utilidad encontraremos módulos compatibles con IIS para ampliar la funcionalidad del servidor web.

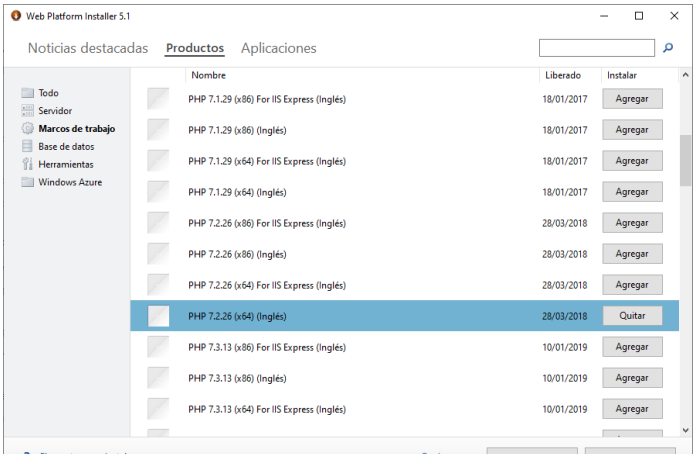

Fig. 11. Instalación de PHP utilizando Web Platform Installer

Una vez finalizada la instalación, ya tendremos nuestro IIS preparado para atender webs dinámicas escritas en PHP.

Podemos encontrar más información sobre cómo instalar y configurar IIS y PHP en el siguiente enlace:

https://learn.microsoft.com/es-es/iis/application-frameworks/scenario-build-a-php-website-on-iis/configuring-step-1-install-iis-and-php

5.3. Instalación de Apache en Ubuntu Server

Ya hemos visto cómo instalar Apache con LAMP de forma rápida y sencilla; ahora vamos a aprender a instalar Apache, MySQL y PHP de forma manual, así como a instalar módulos de PHP.

Los pasos son los siguientes:

Como siempre que instalamos software en Linux, lo primero es actualizar repositorios y paquetes. Para ello ejecutamos:

```
usuario@lnxsrv-web-01:~$ sudo apt-get update && sudo apt-get upgrade
[sudo] password for usuario:
```

Fig. 12. Actualización de repositorios y paquetes https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-quickstart-es

A continuación, instalaremos Apache y el paquete Apache-utils para añadir funcionalidades:

```
usuario@lnxsrv-web-01:~$ sudo apt-get install apache2 apache2-utils
[sudo] password for usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
apache2-utils ya está en su versión más reciente (2.4.29-1ubuntu4.13).
```

Fig. 13. Instalación Apache https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-quickstart-es

Seguiremos con la instalación de la base de datos MySQL. Acto seguido configuraremos la contraseña del usuario **root.**

```
usuario@lnxsrv-web-01:~$ sudo apt-get install mysql-server
usuario@lnxsrv-web-01:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
```

Fig. 14.Instalación y configuración de MySQL – Extracto de https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-

Por último instalaremos PHP. Además del paquete php7.2, instalaremos el módulo de PHP para Apache, y el módulo de conexión con MySQL, que permitirán que el preprocesador PHP pueda interactuar con los diferentes elementos. Instalaremos también el paquete php7.2-common que incluye librerías útiles.

```
usuario@lnxsrv-web-01:~$ sudo apt-get install php7.2 libapache2-mod-php7.2 php7.2-mysql php7.2-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
php7.2-common ya está en su versión más reciente (7.2.24-0ubuntu0.18.04.6).
fijado php7.2-common como instalado manualmente.
php7.2-mysql ya está en su versión más reciente (7.2.24-0ubuntu0.18.04.6).
```

Fig. 15.Instalación de PHP y librerías

Una vez instalados todos los componentes debemos reiniciar Apache para aplicar los cambios y habilitar PHP.

```
usuario@lnxsrv-web-01:~$ sudo apache2ctl restart
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168
.1.149. Set the 'ServerName' directive globally to suppress this message
```

Fig. 16.Reinicio del servicio Apache

En la figura 13 podemos observar que Apache da un aviso en la configuración. Esto es debido a que no tenemos definido un ServerName válido. Podemos solucionarlo añadiendo una línea al final del fichero /etc/apache2/apache. conf, y reiniciando el servicio de Apache.

```
usuario@lnxsrv-web-01:~$ sudo su
root@lnxsrv-web-01:/home/usuario# cat >> /etc/apache2/apache2.conf
serverName localhost
^C
root@lnxsrv-web-01:/home/usuario# apache2ctl restart
```

Fig. 17. Configuración ServerName en Apache

5.4. Instalación de Nginx en Ubuntu Server

En este apartado explicaremos los pasos necesarios para instalar un servidor web con Nginx, PHP y MySQL sobre Ubuntu Server 18.04.

Primero, instalaremos el paquete Nginx y Nginx-common.

```
usuario@lnxsrv-web-01:~$ sudo apt-get install nginx nginx-common
Leyendo lista de paquetes... Hecho
```

Fig. 18. Instalación de Nginx - https://www.digitalocean.com/community/tutorials/como-instalar-nginx-en-ubuntu-18-04-es

A continuación, instalaremos mysql-server y aplicaremos la configuración inicial.

```
usuario@lnxsrv-web-01:~$ sudo apt-get install mysql-server
usuario@lnxsrv-web-01:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
```

Fig. 19.Instalación y configuración de MySQL – Extracto de https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-on-ubuntu-20-04-es

Posteriormente, instalaremos el paquete php-fmp (FastCGI Process Manager)

```
usuario@lnxsrv-web-01:~$ sudo apt-get install php7.2-fpm php7.2-mysql
Leyendo lista de paquetes... Hecho
```

Fig. 20. Instalación de PHP

A diferencia de Apache, en Nginx no es suficiente con reiniciar para activar PHP, necesitaremos habilitarlo manualmente. Para ello editamos el fichero /etc/nginx/sites-available/default

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;


        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
        }

        location ~ \.php$ {
                include snippets/fastcgi-php.conf;

                fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
        }
```

Fig. 21. Instalación Nginx 4 – fichero de configuración de virtualhost

Una vez aplicada la configuración comprobaremos que los cambios son válidos usando el comando nginx –t, y recargando después el servicio para que se active PHP

```
usuario@lnxsrv-web-01:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
usuario@lnxsrv-web-01:~$ sudo service nginx reload
```

Fig. 22. Instalación Nginx 4 – Comprobación de errores en configuración y reinicio servicio

Finalmente, para comprobar el correcto funcionamiento de PHP, crearemos un info.php en nuestro documentRoot.

/ 6. Docker

La aparición de los contenedores ha reinventado la forma de distribución y ejecución de software. Un Docker o contenedor aisla una aplicación y sus dependencias permitiendo desplegar servicios web en cuestión de minutos.

Algunas de las ventajas del uso de contenedores son:

Los contenedores son más eficientes que una máquina virtual. Al no invertir recursos en emular un S.O., los Docker comparten los recursos del host anfitrión.

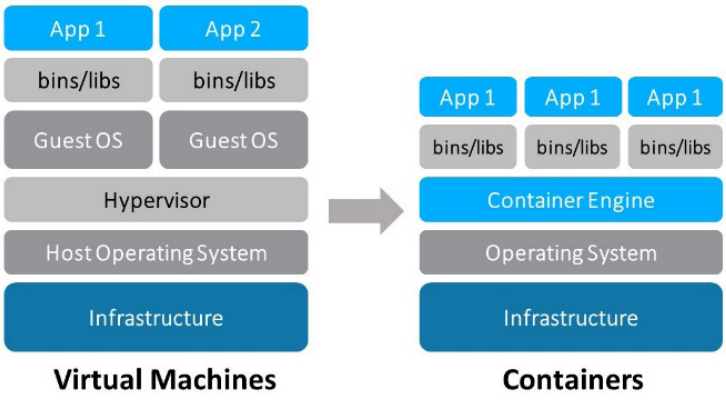

Fig. 23. Estructura docker

Al encapsular las aplicaciones y las librerías necesarias, podremos desplegar el mismo Docker en cualquier equipo, evitando el riesgo de que falten dependencias o existan discrepancias entre distintas versiones de software.

Más ligeros que las máquinas virtuales; con lo que Docker permite la ejecución de múltiples contenedores en el mismo host.

En la figura 24 podemos ver como para 'levantar' un Docker con Apache, se expone el puerto interno 80 al 8080 del servidor docker.

```
$ docker run -dit --name my-running-app -p 8080:80 my-apache2
```

Fig. 24 Ejemplo de un despliegue de un servidor web Apache con Docker

Al tener varios servicios 'corriendo' en contenedores independientes, se mapea el puerto del Docker a un puerto externo del host anfitrión. Este puerto no debe estar siendo utilizado por ningún otro contenedor, o el despliegue fallará.

Para instalar Docker podemos ayudarnos de las guías que proponen en la web oficial de Docker:

https://docs.docker.com/engine/install/

También, en el siguiente vídeo, veremos cómo desplegar un contenedor con Nginx y conoceremos algunos de los comandos utilizados para la gestión de Docker.

Video 2. "Desplegar Nginx con Docker"
https://bit.ly/3ezMa1A

/ 7. Caso práctico 2: “Desplegar Apache con Docker”

Planteamiento: Continuando en el mismo escenario que en el caso 1, compañeros de profesión de Miguel le han comentado las ventajas de trabajar con contenedores; y por tanto, se plantea montar en su empresa el contenedor Docker. No obstante, quiere probarlo primero, y para ello cuenta con un servidor en Ubuntu.

Nudo: ¿Sería posible? ¿Cómo lo haría?

Desenlace: La solución pasaría por instalar Docker en Ubuntu y desplegar un servidor web Apache que escuche por el puerto 8081.

Utilizando como referencia el vídeo 2, podemos instalar Docker y desplegar un contenedor con Apache. Puedes realizar la instalación de Docker utilizando la documentación oficial que encontrarás en la página web:

https://docs.docker.com/engine/install/ubuntu/

Una vez instalado, utiliza el comando “docker search” para localizar la imagen correspondiente.

```
usuario@ubntsrv-docker-01:~$ sudo docker search apache
NAME      DESCRIPTION                    STARS     OFFICIAL
httpd     The Apache HTTP Server Project 3069      [OK]
```

Fig. 25. Búsqueda de imágenes con docker search – Extracto de: https://hub.docker.com/_/httpd

Instala la imagen que acabas de localizar y redirecciona el puerto al 8081.

```
usuario@ubntsrv-docker-01:~$ sudo docker run -d --name Apache -p 8081:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
8559a31e96f4: Already exists
bd517d441028: Pull complete
f67007e59c3c: Pull complete
83c578481926: Pull complete
f3cbcb88690d: Pull complete
Digest:
sha256:387f896f9b6867c7fa543f7d1a686b0ebe777ed13f6f11efc8b94bec743a1e51
Status: Downloaded newer image for httpd:latest
9edddbb60adcddbc0a50587efc1958f6619873dd299aca4f572c0719e6b429ba
usuario@ubntsrv-docker-01:~$
```

Fig. 26. Despliegue de contenedor Apache https://hub.docker.com/_/httpd

Comprueba mediante un navegador web que puedes acceder correctamente.

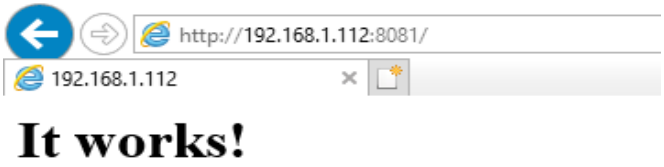

Fig. 27. Prueba de funcionamiento Docker Apache

/ 8. Resumen y resolución del caso práctico de la unidad

En este tema hemos hablado de los **componentes que hacen que un servidor web sea totalmente funcional** y pueda atender peticiones complejas más allá de simples ficheros HTML gracias a PHP y a una base de datos con MySQL.

Posteriormente, **hemos explicado las características de los tres servidores web más conocidos (Apache, Nginx y Microsoft IIS)**, aprendiendo a instalarlos y conociendo sus funcionalidades básicas, hasta llegar a ser capaces de realizar comprobaciones del estado de PHP.

Por último, **nos hemos iniciado en el sistema Docker:** desde su instalación y despliegue básicos, hasta las sentencias más comunes para conocer el estado del servicio.

Resolución del caso práctico de la unidad

En el caso práctico inicial teníamos una situación bastante habitual: Una empresa tiene un servidor web desde hace años y su imagen en internet se ha quedado obsoleta.

Planteamos evaluar todo lo necesario para un 'lavado de cara', y poder transformar su infraestructura e imagen. Para ello, observamos que es necesario cambiar:

- **El servidor:** por antigüedad no soporta actualizaciones de software.
- **Software utilizado:** tiene varias brechas de seguridad que seguramente hayan sido publicadas, y al no tener un parche que las corrija, es un blanco fácil.
- **Web:** La web podría pasar de ser estática escrita en HTML, a un CMS como por ejemplo un WordPress (si no se tienen muchos conocimientos de programación y/o diseño); otorgándole así un mayor atractivo visual y posibilidad de actualizar la web de forma sencilla.

Una de las posibles soluciones a plantear sería migrar los servicios a un servidor Cloud en el que tendremos un servidor web Apache, incluyendo PHP y una base de datos MySQL.

/ 9. Bibliografía

García, J. M. B. (2016, 15 noviembre). Las mejores alternativas a MySQL. Recuperado de https://www.arsys.es/blog/programacion/alternativas-mysql/

Server-side languages · WebPlatform Docs. (s. f.). Recuperado de https://webplatform.github.io/docs/server-side_languages/

Noguera, B. (2014, 22 octubre). ¿Qué es Apache? Recuperado de http://culturacion.com/que-es-apache/

Colaboradores de Wikipedia. (2019, 6 diciembre). Apache Software Foundation. Recuperado de https://es.wikipedia.org/wiki/Apache_Software_Foundation

Instalar LAMP(Linux, Apache, MySQL & PHP) en Ubuntu 17.10. (s. f.). Recuperado de https://ubunlog.com/instalar-lamplinux-apache-mysql-php-en-ubuntu-17-10/

What is a Container? (s. f.). Recuperado de https://www.docker.com/resources/what-container

Mouat, A. (2016). Docker. dpunkt.

Soni, R. (2016). Nginx. Kolkata, West Bengal, India: Apress.