

Cuestiones.

Selecciona una o más de las respuestas posibles:

- 1) El fichero que contiene la configuración del entorno de ejecución de PHP es:
 - a. php.conf.
 - b. php.cfg.
 - c. **php.ini.**
 - d. php.xml.

- 2) El protocolo FTP permite cifrar la información entre cliente y servidor FTP:
 - a. Sí, independientemente del modo de funcionamiento.
 - b. Sí, siempre y cuando cliente sea el origen de la información.
 - c. Sí, mediante AUTH basic auto.
 - d. **No.**

- 3) ¿Qué permisos identifica la combinación rw- r-x r-x?:
 - a. Los permisos 755.
 - b. **Los permisos 655.**
 - c. Los permisos 644.
 - d. Los permisos 744.

- 4) Dos nombres DNS distintos identificando a la misma máquina y a la misma IP, ¿pueden servir una configuración ftp distinta mediante el servidor proFTPD?...
 - a. Sí, es posible configurando virtualhosts.
 - b. No, es imposible para servidores ftp.
 - c. **Sí, es posible, siempre y cuando cada virtualhost y el servidor principal sirvan en un puerto TCP distinto.**
 - d. No, es imposible en cualquier circunstancia.

- 5) Son tipos de cliente FTP:
 - a. **CuteFTP.**
 - b. **Filezilla.**
 - c. **WinSCP.**
 - d. Guszilla.

- 6) Un archivo .jpg se puede transferir como
 - a. Cualquier tipo vale.

b. **Tipo binary.**

c. Tipo ASCII.

d. Al igual que un comprimido .bz2, tipo de archivo binary.

7) Los hosts virtuales basados en IP permiten:

a. **Configurar un dominio por host virtual.**

b. Configurar en una misma IP varios dominios.

c. Configurar en un conjunto de IPs un único dominio local.

d. Todas son correctas.

8) Un archivo WAR no:

a. **Es un conjunto de páginas .css.**

b. Es un conjunto de páginas .html.

c. Permite empaquetar en una sola unidad aplicaciones web de Java.

d. Es un archivo.zip.

9) Para definir el tiempo máximo que puede durar la ejecución de un script PHP se usa la directiva **max_execution_time**.

10) Se emplea el comando **systemctl reload proftpd** para recargar el servicio proFTPD.

(6 puntos).

Desarrollo 1.

Describe, intentando usar tus propias palabras, qué acciones realizan, semejanzas y diferencias de los comandos get, put, mget y mput.

(2 puntos).

Estos comandos se utilizan en el protocolo FTP (File Transfer Protocol) para transferir archivos entre un cliente y un servidor:

- **get:** se utiliza para descargar un archivo específico desde el servidor FTP al cliente. Por ejemplo, si necesitas obtener un archivo llamado archivo.txt del servidor, usarías el comando `get archivo.txt`. Este comando es útil cuando sabes exactamente qué archivo necesitas y solo necesitas transferir un archivo a la vez.
- **put:** se utiliza para subir un archivo específico desde el cliente al servidor FTP. Es el opuesto de get. Por ejemplo, si necesitas subir un archivo llamado archivo.txt al servidor, usarías el comando `put archivo.txt`. Este comando es útil cuando necesitas enviar un archivo específico al servidor.
- **mget:** se utiliza para descargar múltiples archivos desde el servidor FTP al cliente. A diferencia de get, que solo descarga un archivo a la vez, mget permite descargar varios archivos de una sola vez. Puedes

especificar los nombres de los archivos que deseas descargar separados por espacios, o usar comodines como * para seleccionar múltiples archivos que coincidan con un patrón. Por ejemplo, mget *.txt descargaría todos los archivos con extensión .txt del servidor.

- **mput:** se utiliza para subir múltiples archivos desde el cliente al servidor FTP. Es el opuesto de mget. Permite subir varios archivos de una sola vez, similar a cómo mget descarga múltiples archivos. Puedes especificar los nombres de los archivos que deseas subir separados por espacios, o usar comodines como * para seleccionar múltiples archivos que coincidan con un patrón. Por ejemplo, mput *.txt subiría todos los archivos con extensión .txt del cliente al servidor.

Semejanzas:

- **Transferencia de archivos:** Tanto get y put como mget y mput se utilizan para transferir archivos entre el cliente y el servidor FTP.
- **Comodines:** mget y mput permiten el uso de comodines para seleccionar múltiples archivos, lo que facilita la transferencia de varios archivos a la vez.

Diferencias:

- **Número de archivos:** get y put se utilizan para transferir un solo archivo a la vez, mientras que mget y mput se utilizan para transferir múltiples archivos a la vez.
- **Dirección de transferencia:** get y mget se utilizan para descargar archivos del servidor al cliente, mientras que put y mput se utilizan para subir archivos del cliente al servidor.

Desarrollo 2.

Ventajas e inconvenientes de contenedores de servlets **dentro-de-proceso** versus contenedores de servlets **fuera-de-proceso** y en qué casos preferirías unos sobre otros y por qué.
(2 puntos).

Los contenedores de servlets son esenciales en el desarrollo de aplicaciones web en Java. Pueden funcionar de dos maneras principales: dentro-de-proceso (in-process) y fuera-de-proceso (out-of-process):

1) Contenedores de Servlets Dentro-de-Proceso (In-Process)

Ventajas:

- **Integración Directa:** Los contenedores dentro-de-proceso se integran directamente con el servidor web, lo que simplifica la configuración y el despliegue.
- **Rendimiento:** Pueden ofrecer mejor rendimiento ya que no hay necesidad

de redirigir las solicitudes entre diferentes procesos.

- **Facilidad de Uso:** Son más fáciles de configurar y usar, especialmente para proyectos pequeños y medianos.

Inconvenientes:

- **Limitación de Recursos:** Pueden consumir más recursos del servidor web, lo que puede afectar el rendimiento general.
- **Menor Flexibilidad:** La integración directa puede limitar la flexibilidad en la elección de diferentes servidores web o contenedores.

2) Contenedores de Servlets Fuera-de-Proceso (Out-of-Process)

Ventajas:

- **Aislamiento:** Los contenedores fuera-de-proceso se ejecutan en procesos separados, lo que proporciona un aislamiento mejor y reduce el impacto en el servidor web.
- **Flexibilidad:** Ofrecen mayor flexibilidad en la elección de diferentes servidores web y contenedores, permitiendo una configuración más personalizada.
- **Escalabilidad:** Son más fáciles de escalar, ya que se pueden agregar más instancias de contenedores sin afectar el servidor web principal.

Inconvenientes:

- **Configuración Compleja:** Requieren una configuración más compleja, incluyendo la configuración de redirecciones y balanceo de carga.
- **Latencia:** Pueden introducir una latencia adicional debido a la necesidad de redirigir las solicitudes entre procesos.

3) Casos de Uso

Preferir Contenedores Dentro-de-Proceso:

- **Proyectos Pequeños y Medianos:** Para proyectos que no requieren una alta escalabilidad o aislamiento, los contenedores dentro-de-proceso son una opción más sencilla y eficiente.
- **Desarrollo Local:** Durante el desarrollo local, la facilidad de configuración y el rendimiento mejorado pueden ser beneficios significativos.

Preferir Contenedores Fuera-de-Proceso:

- **Proyectos Grandes y Complejos:** Para proyectos que requieren alta escalabilidad, aislamiento y flexibilidad, los contenedores fuera-de-proceso son más adecuados.
- **Entornos de Producción:** En entornos de producción, donde la estabilidad y el rendimiento son cruciales, la capacidad de escalar y aislar los contenedores puede ser esencial.