



# T1 Tema 1: LAMP

## ¿Qué aprenderás?

- Instalar y configurar una plataforma LAMP
- Instalar y configurar Tomcat

## ¿Sabías que...?

- LAMP es el acrónimo de Linux, Apache, MySQL y PHP
- MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, seguramente el más usado en aplicaciones creadas como software libre.



## 1. Servidores de aplicaciones

---

Un "servidor de aplicaciones" es una denominación muy genérica de un sistema, es muy poco concreta. Por servidor de aplicaciones se podría entender cualquier servidor que ejecuta aplicaciones para los clientes que se lo solicitan. El servidor ejecuta el programa y les envía a los clientes los resultados, de esa forma, toda la carga de proceso de la ejecución la asume el servidor, y libera a los clientes de ella. Hoy en día pueden existir diversas tecnologías asociadas a esta denominación.

Estos apuntes concretan más en este tema, refiriéndose en todo momento a un "servidor de aplicaciones web". Con este término, se concreta el hecho de que los clientes accedan vía web (http) al servidor, y le soliciten la ejecución de algún código. El servidor lo ejecutará y los resultados se los enviará a los clientes también vía web, es decir, les dará el formato adecuado y los enviará mediante http, por ejemplo, y los hará llegar a los clientes, cuyos navegadores presentarán los resultados de la ejecución.

Cada vez que el cliente accede al servidor de aplicaciones web, éste ejecuta un código y produce un resultado que puede variar en cada acceso. La página web resultante que ve el cliente no es siempre la misma, de alguna forma se trata de una página web dinámica.

Sin ir más lejos, cuando se accede a la página web del banco para consultar el extracto de la cuenta bancaria, quien atiende a todos los clientes que lo solicitan es un servidor (o muchos servidores) de aplicaciones web, el cual, cada vez que se recibe la petición, ejecuta un código que le hace acceder a una base de datos a recoger unos datos concretos (movimientos de la cuenta), y presentarlos en formato html filtrados según haya decidido el cliente.

Un servidor web Apache con un contenido html estático no es suficiente para considerarse un servidor de aplicaciones web. Será necesario añadir tecnologías que ejecuten un código cuando llegue la petición web del cliente.

También el término "servidor de aplicaciones web" se puede llevar a la práctica de diferentes maneras y con diferentes tecnologías, así que también sería necesario concretarlo.

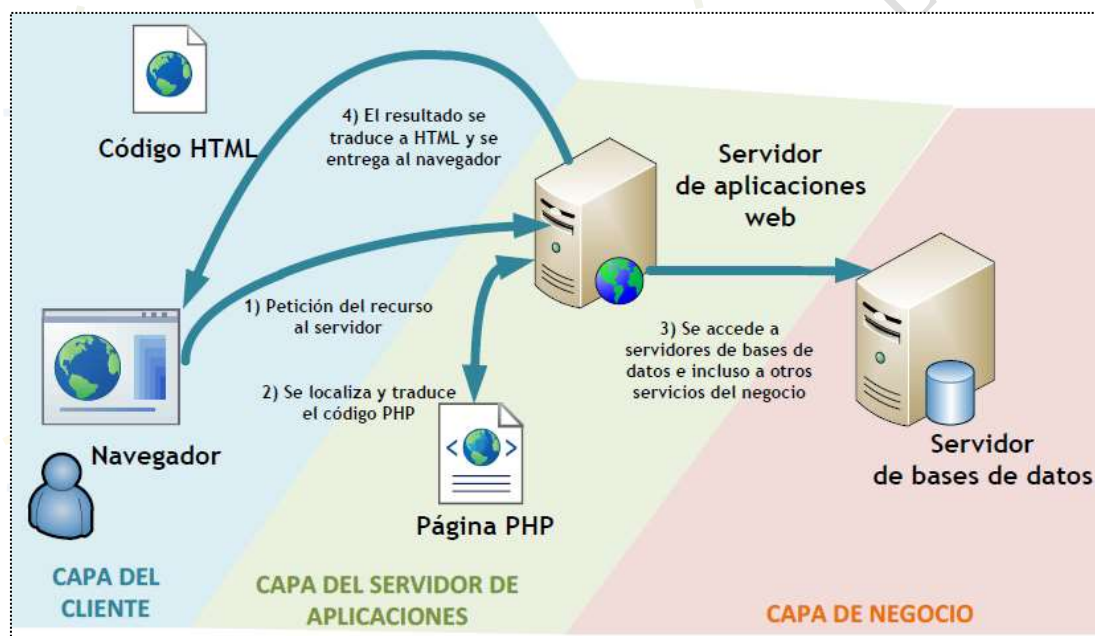


## 2. LAMP

Un primer ejemplo de servidor de aplicaciones web podría ser un servidor Apache, con los módulos correspondientes para la ejecución de código PHP instalados. Con esos módulos, el servidor puede procesar el código PHP, que está incrustado en el contenido html. Después de su ejecución, se envía todo el resultado html a los clientes vía http. Los clientes no ejecutan nada en este caso, tan sólo hacen la petición y reciben el resultado de la ejecución ya en formato html.

Es muy común que los servidores de aplicaciones web accedan a bases de datos para consultar o escribir datos almacenados. Esas bases de datos pueden estar en otros sistemas comunicados a través de redes TCP/IP locales.

En la imagen de ejemplo siguiente se muestra el proceso comentado anteriormente:



La página con código PHP es interpretada por el servidor y traducida a html con los resultados de la ejecución. Apache requiere de la instalación de módulos adicionales para poder interpretar el código PHP. El resultado final se envía al cliente que ya no tiene que procesar nada, tan sólo el navegador se encargará de mostrar el contenido recibido. Toda la ejecución ha sucedido en el servidor.

En la imagen anterior se muestra el esquema dividido en tres capas. Esta división es conocida como un modelo de tres capas, donde están claramente diferenciadas las funciones de cada una:

- Capa de presentación: la que ve el usuario, es la interfaz gráfica a través de la cual el usuario interactúa con la aplicación.



- Capa de aplicación: donde residen los programas que se ejecutan. Se reciben las peticiones, y se envían los resultados del proceso.
- Capa de datos: donde están almacenados los datos.

Aunque todas las capas pueden estar en una misma máquina, por razones de seguridad y escalabilidad, es común que se separen en sistemas diferentes. Cada capa tiene sus propias necesidades específicas, y estar instaladas por separado ayuda a una mayor flexibilidad y eficiencia del sistema global.

El modelo de tres capas se despliega a menudo juntando unas tecnologías y productos concretos. La mayor parte de los servidores de aplicaciones web desplegados en Internet incluyen elementos de LAMP. LAMP es un conjunto de productos que trabajando juntos han tenido éxito y mucha fiabilidad.

LAMP es acrónimo de Linux, Apache, MySQL y PHP. Linux es el sistema operativo, Apache el servidor Web, MySQL el gestor de bases de datos, y PHP el lenguaje de programación (también puede incluir a otros como Perl o Python). Aunque no es un estándar, es un conjunto utilizado mayoritariamente.



El modelo de tres capas mostrado hace referencia al modelo de los sistemas implicados en una aplicación web. A menudo se relaciona también este modelo con el modelo de arquitectura del software "modelo-vista-controlador", aunque este último queda fuera del alcance de estos apuntes.

## 1.1. Instalación de una plataforma LAMP

A continuación se muestran los procesos a seguir para instalar una plataforma LAMP con servidor **Linux Ubuntu 18.04**. Se instalará Apache con los módulos correspondientes para la interpretación de código PHP, también se instalará un gestor de bases de datos MySQL.

Mediante un pequeño código de ejemplo en PHP se realizará un acceso simple a una base de datos existente en el servidor de bases de datos MySQL.

### 1.1.1. Instalación de módulos PHP en Apache

Una vez instalado el servidor web Apache en una máquina, para que éste pueda interpretar código PHP se debe instalar el paquete de php para el sistema operativo y el



módulo de apache correspondiente. En concreto se deben instalar esos dos componentes mediante:

```
sudo apt install php libapache2-mod-php
```

La manera más rápida de comprobar el correcto funcionamiento de los paquetes instalados es ejecutar un código php de test, con la función phpinfo(). Se debe crear un archivo con el siguiente contenido:

```
<?php
    phpinfo();
?>
```



El archivo se debe ubicar en el DocumentRoot del servidor apache (*/var/www/html*) con el nombre que se desee. Si el archivo se llama "phpinfo.php", por ejemplo, ya que se encuentra en el document root, se podrá acceder vía web desde un cliente a la siguiente dirección:

```
http://hostname/phpinfo.php
```

Donde "hostname" se puede sustituir por la IP o nombre del servidor apache.

Si todo ha ido bien, la función phpinfo mostrará mucha información sobre el sistema:



PHP Version 7.2.10-0ubuntu0.18.04.1	
	
System	Linux (linka-server 4.15.0-35-generic) #39-Ubuntu SMP Mon Sep 24 16:19:09 UTC 2018 x86_64
Build Date	Sep 13 2018 13:45:02
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/15-xm.ini, /etc/php/7.2/apache2/conf.d/20-bz2.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mysql.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-wddx.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib*, string.rot13, string.rot48, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2*, convert.conv.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies with Zend OPcache v7.2.10-0ubuntu0.18.04.1, Copyright (c) 1999-2018, by Zend Technologies	
	

Si la información se visualiza correctamente, Apache ya puede interpretar correctamente PHP.

Para poder hacer uso de funciones de php que accedan a una base de datos Mysql también se deben instalar unos módulos adicionales. Al instalar estos módulos de PHP, se podrá hacer uso en el código PHP de funciones como:

- mysql\_connect
- mysql\_error
- mysql\_select\_db

Con éstas y muchas otras, se podrá acceder a bases de datos desde el código PHP.

Para instalar el módulo correspondiente en el servidor apache:

```
sudo apt install php-mysql
```



### 1.1.2. Instalación de MySQL

MySQL es un gestor de bases de datos relacionales. Es código abierto y actualmente es propiedad de Oracle. Hoy en día es el gestor de BBDD más utilizado en el entorno de las aplicaciones web y se considera una pieza clave de la plataforma LAMP.

A continuación se muestran los pasos para instalar un sistema gestor de bases de datos como MySQL.

Para instalar el servicio mediante apt:

```
sudo apt install mysql-server
```

El paquete instalado incluye tanto el servidor MySQL como la línea de comandos cliente para manejarlo.

Cuando la instalación esté completa, se debe ejecutar un archivo de comandos de seguridad que viene preinstalado con MySQL, éste removerá algunos parámetros peligrosos, así como asegurará el acceso a la base de datos.

Ejecuta el archivo interactivo de comandos mediante:

```
sudo mysql_secure_installation
```

Preguntará si se quiere configurar el conector de validación de contraseña: VALIDATE PASSWORD PLUGIN.

Responde Y si se está de acuerdo, cualquier otra respuesta continuará sin realizar la habilitación.

```
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?
```

```
Press y|Y for Yes, any other key for No:
```

Si se responde "yes", solicitará que se seleccione el nivel de validación de contraseña. Se debe tener en cuenta que si se selecciona 2 representando el nivel más fuerte, se recibirá errores al intentar utilizar una contraseña que no contenga números, letras mayúsculas y minúsculas, así como caracteres especiales; además la contraseña no podrá estar basada en palabras comunes en un diccionario.





```
There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1
```

Sin importar el nivel escogido para VALIDATE PASSWORD PLUGIN, el servidor solicitará, a continuación, seleccionar y confirmar la contraseña para el usuario root de MySQL. Ésta es una cuenta administrativa dentro MySQL con privilegios incrementados. Puede ser entendida de manera similar a la cuenta root del servidor mismo (Sin embargo, estarás configurando una cuenta específica para MySQL). Hay que asegurarse de utilizar una contraseña fuerte y única, no debería dejarse en blanco.

Si se habilitó la validación de contraseña, se mostrará qué tan fuerte es la contraseña para la cuenta root que se acaba de introducir y el servidor preguntará si se quiere cambiar. Si se cree que es adecuado como está, se seleccionará N para seleccionar "no" en la línea de comandos:

```
Using existing password for root.

Estimated strength of the password: 100
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
```

Para las siguientes preguntas, se presionará Y y se pulsará la tecla Enter en cada sugerencia. Esto removerá algunos usuarios anónimos y la base de datos de prueba, deshabilitará ingresos remotos del root, y cargará estas nuevas reglas, de tal modo que MySQL respete inmediatamente los cambios que se acaban de hacer.

Una vez instalado se puede comprobar que el servicio está en marcha mediante:

```
sudo netstat -tap | grep mysql
```

Un ejemplo de resultado del comando de supervisión ejecutado justo después de instalar el servicio:

```
linkia@linkia-server:/$ sudo netstat -tap | grep mysql
tcp        0      0 linkia-server:mysql  0.0.0.0:*        LISTEN      3690/mysql
```

También se puede comprobar con "netstat -nat". En este caso se muestran las IPs y puertos numéricos en vez de nombres descriptivos:

```
linkia@linkia-server:/$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
```



Se comprueba como el puerto típico de MySQL, el 3306 está a la escucha.

Con esta instalación el sistema gestor de BBDD ya está en marcha pero aún no es capaz de dar servicio a través de la red. En este estado inicial el servidor tan solo es accesible desde sí mismo (127.0.0.1), y no desde clientes externos. Para habilitar el acceso desde la red se deberá editar el fichero de configuración general del servicio.

El fichero a editar es en concreto:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

En dicho fichero se debe encontrar el parámetro "bind-address":

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
#
```

Como se observa, el servicio escucha tan solo a través de la dirección de localhost (127.0.0.1). Ese valor se debe editar y poner el valor de la IP propia del servidor MySQL. Por ejemplo, si el servidor tiene la IP 172.16.0.104, el parámetro quedaría:

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 172.16.0.104
#
```

Una vez hecho el cambio se debe reiniciar el servicio. Obsérvese en la imagen siguiente cómo después de reiniciar el servicio, al listar el estado del servicio, ahora se indica que está escuchando en la IP del servidor (en el ejemplo 172.16.0.104).

```
linkia@linkia-server:/$ service mysql restart
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'mysql.service'.
Authenticating as: linkia
Password:
==== AUTHENTICATION COMPLETE ====
linkia@linkia-server:/$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address          State
tcp        0      0 127.0.0.53:53           0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN
tcp        0      0 172.16.0.104:3306       0.0.0.0:*                LISTEN
```

Nótese las diferencias respecto a la ejecución del comando netstat mostrada anteriormente.

A continuación se comprobará el acceso al gestor de BBDD mediante el cliente de línea de comandos instalado en el propio servidor. El acceso se realizará mediante el comando "mysql -u root -p":



```
linkia@linkia-server:/$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.23-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Tras poner el password del root, establecido en el momento de la instalación, se accede a la línea de comandos de control del sistema.

Con los siguientes comandos de ejemplo, se procede a la creación de una base de datos de nombre testdb, se crea un usuario de nombre testuser (con password Qwerty321), al cual se le dan privilegios totales en la base de datos creada:

```
CREATE DATABASE testdb;
SHOW DATABASES;
CREATE USER 'testuser'@'%' IDENTIFIED BY 'Qwerty321';
GRANT ALL PRIVILEGES ON testdb.* TO 'testuser'@'%';
```

Nótese como al final de cada comando se debe incluir el símbolo de punto y coma.

```
mysql> CREATE DATABASE testdb;
Query OK, 1 row affected (0,01 sec)

mysql> CREATE USER 'testuser'@'%' IDENTIFIED BY 'Qwerty321';
Query OK, 0 rows affected (0,00 sec)

mysql> GRANT ALL PRIVILEGES ON testdb.* TO 'testuser'@'%';
Query OK, 0 rows affected (0,00 sec)
```

El comando SHOW DATABASES es de mera comprobación de la creación de la base de datos. Si se ejecuta, se observa la nueva base de datos junto a las bases de datos de sistema:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| testdb |
+-----+
4 rows in set (0.05 sec)
```



Como comprobación final, se hará una prueba de conectividad desde una máquina externa a través de la red. Para ello, la máquina remota tendrá que tener instalado el cliente mysql. La instalación del cliente mediante apt se realiza con el commando: "apt install mysql-client".

Una vez instalado, se muestra un ejemplo de conexión desde un cliente contra un servidor cuya IP es 172.16.0.104:

```
linkia@linkia-pc:~$ mysql -h 172.16.0.104 -u testuser -p testdb
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.23-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

La opción -h indica la IP del servidor al que se desea conectar, con las opciones -u y -p se indica el usuario con el que se desea realizar la conexión (el password se pide de forma interactiva). En la parte final del comando se debe indicar el nombre de la base de datos a la que se desea acceder.

Con este último comando se comprueba el correcto funcionamiento del servidor MySQL instalado y la base de datos creada.

### 2.1.1. Gestión web de MySQL mediante phpmyadmin

La gestión del servidor MySQL mediante comandos puede resultar muy tediosa. Para ayudar, existen diversas interfaces web programadas para controlar por completo MySQL.

A través de la interfaz web se pueden realizar todas las acciones necesarias con las bases de datos y con el servidor. Una de las interfaces más populares es phpmyadmin. Esta interfaz está programada en PHP y requiere de un servidor web para instalarse.

En este caso, se instalará phpmyadmin en el mismo servidor MySQL. Al instalar el paquete correspondiente también se instalará un apache para soportarlo, junto con los módulos de PHP necesarios.

Previo al comando de instalación, se tiene que actualizar al sistema y añadir los servidores fuente en los repositorios:

```
sudo apt update
sudo apt upgrade
```



```
sudo add-apt-repository universe
```

El comando de instalación es:

```
sudo apt install phpmyadmin
```

Durante la instalación, se requiere elegir cuál será el servidor web a configurar por parte de phpmyadmin. En este caso será apache2:

Configuración de phpmyadmin

Por favor, elija el servidor web que se debería configurar automáticamente para que ejecute phpMyAdmin.

Servidor web que desea reconfigurar automáticamente:

☒ apache2  
☐ lighttpd

<Aceptar>

Para la configuración de la propia interfaz phpmyadmin se requiere una base de datos. En el proceso de instalación se puede decidir que se cree automáticamente, con la opción "Sí"

Configuración de phpmyadmin

Es necesario tener una base de datos instalada y configurada para phpmyadmin antes de poder utilizarlo. Puede gestionar esto opcionalmente a través «dbconfig-common».

Si vd. es un administrador de bases de datos avanzado o si la base de datos ya está instalada y configurada quizás quiera realizar esta configuración manualmente, y debería rechazar esta opción. Probablemente podrá encontrar los detalles de las operaciones que debe realizar en «/usr/share/doc/phpmyadmin».

Debería escoger esta opción en cualquier otro caso.

¿Desea configurar la base de datos para phpmyadmin con «dbconfig-common»?

☒ Sí ☐ No

Para que se pueda crear la base de datos, se deberán dar las credenciales (P@ssword1234) del usuario root de MySQL

Configuración de phpmyadmin

Proporcione la contraseña para la cuenta de administración que este paquete utilizará para crear la base de datos MySQL y el usuario.

Contraseña del usuario de administración de la base de datos:

\_\_\_\_\_

<Aceptar> <Cancelar>



El servidor apache instalado quedará listo para servir la página de phpmyadmin, que dará acceso a la gestión del servidor MySQL. En este ejemplo, todos esos componentes quedarán instalados en una misma máquina, aunque podría haberse separado entre máquinas diferentes.

Para acceder a phpmyadmin se requiere un navegador web. Desde cualquier máquina con conectividad con el servidor se podrá acceder mediante la URL:

<http://servername/phpmyadmin>

Donde "servername" debe ser el nombre o dirección IP del servidor donde se encuentre instalado phpmyadmin.

Por ejemplo, si el servidor tiene la IP 172.16.0.104, se muestra un ejemplo de acceso:

phpMyAdmin

Bienvenido a phpMyAdmin

Idioma - Language

Español - Spanish

Iniciar sesión

Usuario:

Contraseña:

Continuar

El servicio exige que el usuario se identifique para acceder al gestor de BBDD. Al acceder como usuario root se pueden observar todas las bases de datos del sistema, incluida la que se creó anteriormente mediante línea de comandos (testdb en el ejemplo)





Se puede observar como el entorno de trabajo es mucho más amigable que la línea de comandos tradicional.

Como se ha comentado anteriormente, phpmyadmin no tiene por qué estar instalado en la misma máquina que el servidor MySQL que va a administrar. De hecho, con la misma interfaz phpmyadmin se podrían administrar diferentes servidores. Si se instala phpmyadmin en un servidor diferente, posteriormente se le debe indicar donde se encuentran (en qué IPs) los servidores mysql a administrar.

Esta configuración se realiza en el fichero:

`/etc/phpmyadmin/config.inc.php`

En concreto, se debe modificar el valor "localhost" de la sección que se muestra a continuación del fichero indicado:

```
/* Configure according to dbconfig-common if enabled */
if (!empty($dbname)) {
    /* authentication type */
    $cfg['Servers'][$i]['auth_type'] = 'cookie';
    /* Server parameters */
    if (empty($dbserver)) $dbserver = 'localhost';
    $cfg['Servers'][$i]['host'] = $dbserver;
}
```

Se debe sustituir el valor "localhost" por la IP del servidor que se desea administrar.

### 2.1.2. Acceso a base de datos con PHP

A modo de ejemplo se presenta un código PHP básico que realiza una conexión contra una base de datos ubicada en una máquina remota.

En el ejemplo, el servidor de bases de datos tiene la IP 172.16.0.104, y se programa una conexión a la base de datos de nombre "testdb", y con el usuario "testuser".



```
<HTML><BODY><H1>Prueba de acceso a BBDD</H1>
<?php
$servername = "172.16.0.104";
$username = "testuser";
$password = "Qwerty321";
$dbname = "testdb";

//Crear la conexión
$conn = mysqli_connect($servername, $username, $password, $dbname);

//Probar la conexión
if(!$conn) {
die("Error en la conexión: " . mysqli_connect_error());
}

echo "Conexion ok!!";

?>
</BODY></HTML>
```

El código PHP está incrustado en el código html, y será interpretado por apache gracias a los módulos php instalados.

Si el fichero se almacena con el nombre testdb.php, se deberá acceder al servidor web mediante la IP del servidor web e indicando el nombre del fichero:

<http://172.16.0.104/testdb.php>

El servidor web tiene la IP 172.16.0.104 y el código PHP conecta al servidor web con el servidor MySQL, que tiene la IP 172.16.0.104.

A continuación se muestra un ejemplo de acceso al servidor web desde un cliente externo:



Otros ejemplos de cómo programar el acceso a bases de datos con código PHP están en:

<http://www.w3schools.com/php>

A continuación se muestran ejemplos de operaciones básicas con bases de datos. En todos ellos, al principio del código, se deben dar los valores adecuados para las variables de usuario, password, servidor y base de datos.





Ej. Creación de base de datos:

#### Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";

if (mysqli_query($conn, $sql)) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Ej. Inserción de datos

#### Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```



### Ej. Select

#### Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

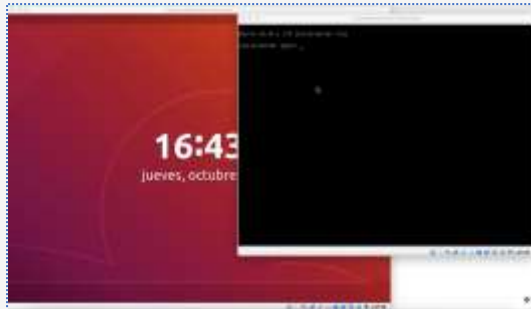
mysqli_close($conn);
?>
```



## Recursos/Enlaces

---

- Vídeo instalación LAMP



- Ubuntu server [Enlace]

## Conceptos clave

---

- **Apache:** servidor web HTTP de código abierto.
- **PHP:** es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.
- **MySQL:** es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL). MySQL se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows.
- **PHPMyAdmin:** es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.



## Test de autoevaluación

---

Pon el significado de cada una de las siglas de LAMP:

¿Cómo se llama el usuario creado en el gestor de bases de datos MySQL utilizando el comando CREATE USER en estos apuntes?

- a. pruebas
- b. user
- c. testuser
- d. usertest

¿En qué ubicación se encuentra el DocumentRoot del servidor Apache?

- a. /var/www/html
- b. /var/htdocs/html
- c. /www/htdocs/html
- d. /html/docs



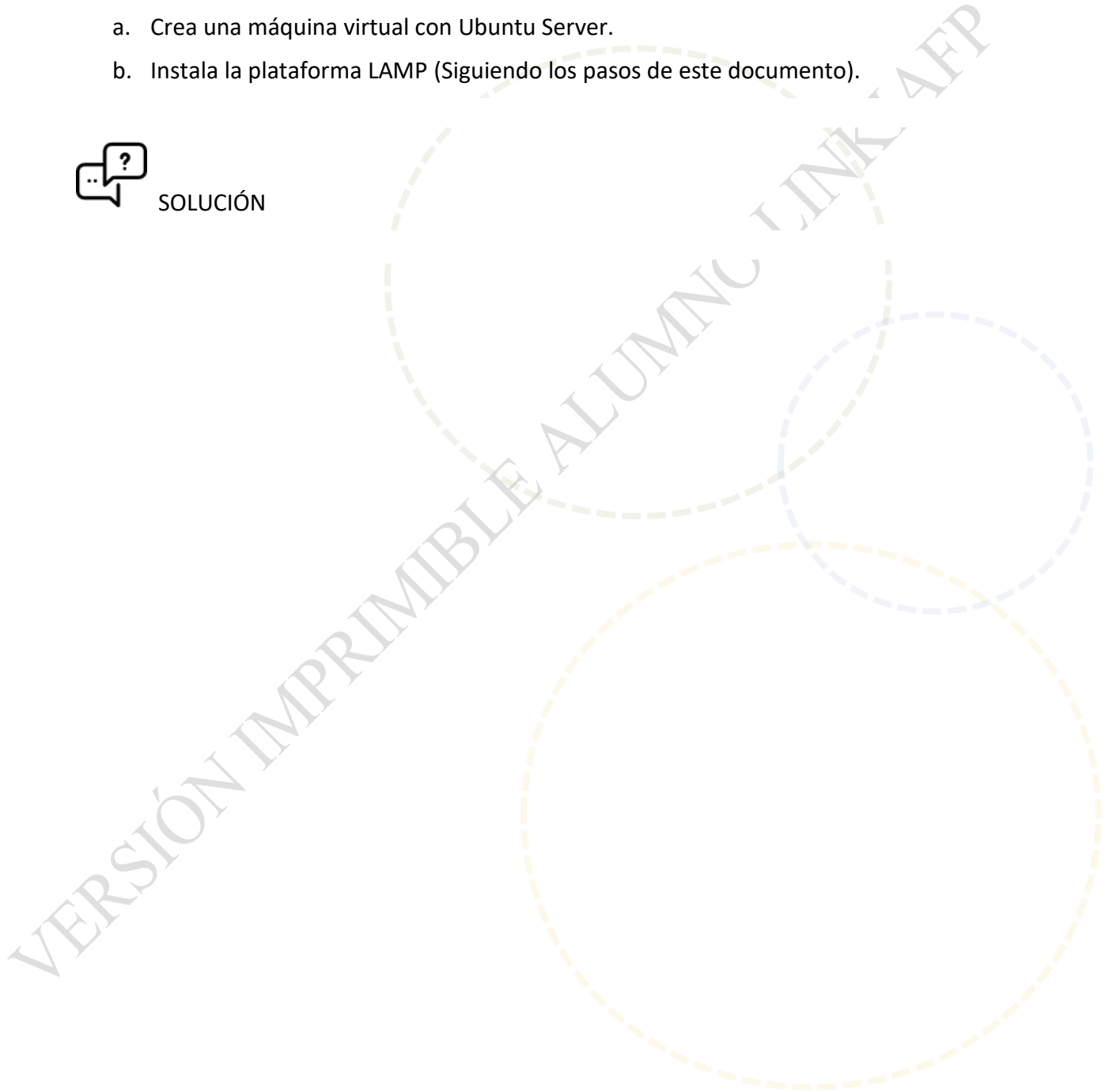
## Ponlo en práctica

### Actividad 1

1. Instalación de LAMP:
  - a. Crea una máquina virtual con Ubuntu Server.
  - b. Instala la plataforma LAMP (Siguiendo los pasos de este documento).



SOLUCIÓN



# Tema 2: Tomcat

## ¿Qué aprenderás?

---

- Instalar un servidor Tomcat.
- Configurar el acceso al servidor Tomcat a través de la interfaz web.
- Desplegar una web en un servidor Tomcat.

## ¿Sabías que...?

---

- El creador de Tomcat eligió como icono un gato pretendiendo representar la capacidad de cuidarse de sí mismo y ser independiente que tienen los gatos.
- Un archivo WAR (de Web Application Archive - Archivo de aplicación web) es un archivo JAR utilizado para distribuir una colección de JavaServer Pages, servlets, clases Java, archivos XML, bibliotecas de tags y páginas web estáticas (HTML y archivos relacionados) que juntos constituyen una aplicación web.



## 1. Tomcat

Otro ejemplo de servidor de aplicaciones web puede ser Tomcat. Tomcat es un producto de la fundación Apache, que es capaz de ejecutar código Java y presentar los resultados en formato web html para los clientes. Aunque Tomcat puede hacer también las funciones de servidor web por sí solo, es habitual que funcione haciendo equipo con Apache como servidor web, mientras Tomcat se dedica exclusivamente a la ejecución del código Java.

Estrictamente hablando, Tomcat no es un servidor de aplicaciones, es un contenedor de servlets, que implementa las especificaciones Java Servlet y JavaServer Pages (JSP) de Sun Microsystems (actualmente Oracle).

Un servlet es un programa Java que extiende las capacidades de un servidor. Habitualmente implementa aplicaciones albergadas en servidores web. Un servlet es código Java que produce resultados en formato html, para que después el servidor web pueda enviar al cliente mediante http.

A continuación se muestra un ejemplo de servlet:

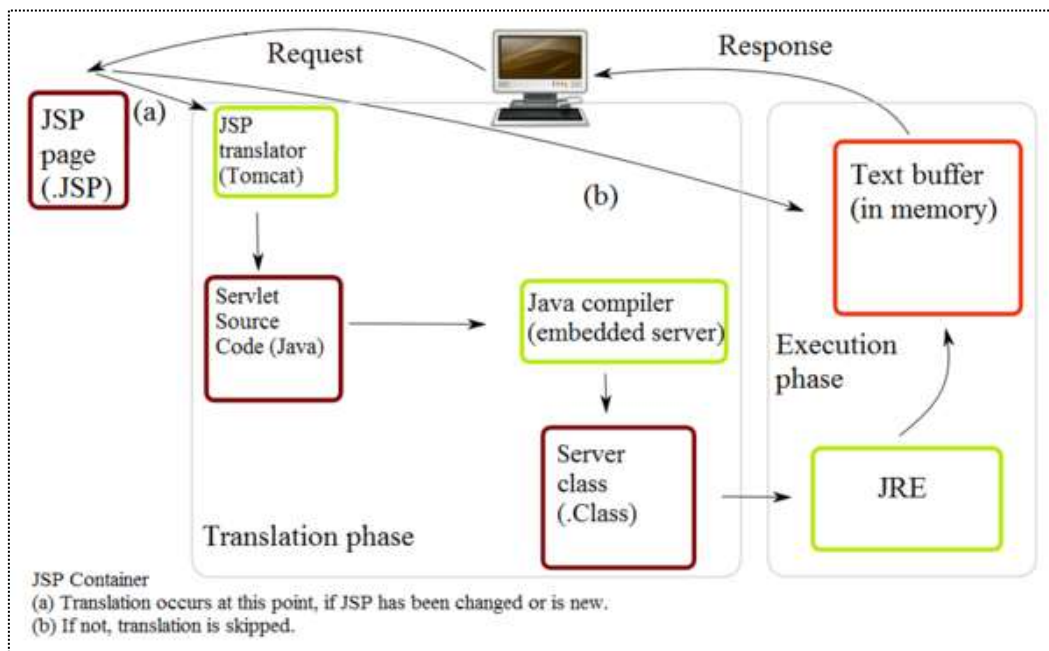
```
1  public class HelloServlet extends HttpServlet {  
2      public void doGet(HttpServletRequest request, HttpServletResponse response) {  
3          response.setContentType("text/html");  
4          PrintWriter out = response.getWriter();  
5          out.println("<title>Hello World!</title>");  
6      }  
7      ...  
8  }
```

Se puede observar cómo los resultados del programa en Java están formateados con html.

La tecnología de servlets produce grandes ventajas respecto a las tecnologías tradicionales CGI que permitían ejecutar código de diversos tipos en servidores web. Antiguamente, con CGI, un nuevo proceso se generaba por cada petición HTTP. Con servlets, la Java Virtual Machine está siempre en marcha y gestiona cada petición con mucha más eficiencia.

Por otro lado, JSP (Java Server Pages) son una abstracción de alto nivel de los servlets. Mediante JSP se puede escribir código Java (Java-like) dentro de páginas html. Cuando el servidor procesa la página, el JSP es traducido a un servlet para su ejecución en el servidor.

Este proceso de traducción se realiza de forma inteligente. Para la primera petición recibida de un cliente, se debe realizar todo el proceso de traducción, compilación y ejecución, pero para las siguientes peticiones sólo es necesaria la ejecución ya que se almacena el resultado previo en memoria buffer:



wikipedia.org

Si un servlet se puede entender como código html dentro de código Java, un JSP se puede entender como código java dentro del código html.

A continuación se muestra un ejemplo de JSP:

```

Start Page x index.jsp x
1  <%--
2      Document    : index
3      Created on  : 02-Jul-2009, 23:05:22
4      Author     : Siriquelle
5  --%>
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
9  "http://www.w3.org/TR/html4/loose.dtd">
10 <%
11     String h = "hello";
12     String w = "world";
13 %>
14 <html>
15     <head>
16         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
17         <title>JSP Page</title>
18     </head>
19     <body>
20         <h1>
21             <%= h + w + "!!" %>
22         </h1>
23     </body>
24 </html>
25

```

<http://threadbarecanvas.azurewebsites.net/java-web/java-web-hello-world/>





Como ya se ha mencionado, Tomcat no es un servidor de aplicaciones web por sí solo. Un servidor de aplicaciones tiene muchas otras funcionalidades como caché, clustering, alta disponibilidad, seguridad, monitorización, etc.

Productos como JBoss (WildFly) o GlassFish encajan mejor en la definición de servidor de aplicaciones web, aunque quedan fuera del alcance de estos apuntes.

## 1.1. Instalación de Tomcat

A continuación se muestra el procedimiento para la instalación de Tomcat en un sistema operativo Linux Ubuntu Server.

### 1.1.1. Instalación de Java

Tomcat requiere la máquina virtual de Java en el sistema donde se instale. Estrictamente solo es necesario el JRE (Java Running Engine), aunque se instalará el JDK, que incluye el JRE además de las herramientas de desarrollo de Java.

Se puede elegir entre instalar la versión propietaria de Oracle o la versión open-source. Ambas son adecuadas para la instalación posterior de Tomcat. En este caso se procederá a la instalación de la versión propietaria Oracle.

Dado que la versión propietaria de Oracle no está disponible en los repositorios de Ubuntu, se debe añadir antes el repositorio correspondiente:

```
sudo add-apt-repository ppa:linuxuprising/java
```

Con el comando anterior se añade un repositorio adicional, donde se encuentra Java.

```
linkia@linkia-server:~$ sudo add-apt-repository ppa:linuxuprising/java
Oracle Java 10 installer

Java binaries are not hosted in this PPA due to licensing. The packages in this PPA download and install Oracle Java 10 (JDK 10), so a working Internet connection is required.

The packages in this PPA are based on the WebUpd8 Oracle Java PPA packages: https://launchpad.net/~webupd8team/+archive/ubuntu/java

Created for users of https://www.linuxuprising.com/

Issues or suggestions? Leave a comment here:
Oracle Java 10: https://www.linuxuprising.com/2018/04/install-oracle-java-10-in-ubuntu-or.html
Oracle Java 11: https://www.linuxuprising.com/2018/10/how-to-install-oracle-java-11-in-ubuntu.html
More info: https://launchpad.net/~linuxuprising/+archive/ubuntu/java
Press [ENTER] to continue or Ctrl-c to cancel adding it.
```



En esta ocasión, el repositorio informa de las diferentes versiones de Java disponibles.

Después de añadir el repositorio, se realiza el "apt-get update", que esta vez actualizará también el contenido del repositorio añadido.

```
sudo apt update
```

A continuación, se puede instalar Java:

```
sudo apt install oracle-java11-installer
```

Una vez instalado, con el comando "java -version" se puede comprobar que la instalación ha acabado bien, y java está en marcha:

```
linkia@linkia-server:~$ java -version
java version "11" 2018-09-25
Java(TM) SE Runtime Environment 18.9 (build 11+28)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11+28, mixed mode)
```

Antes de empezar a instalar Tomcat es necesario configurar las variables de entorno del sistema. Las variables de entorno son consultadas por Tomcat para localizar los directorios donde Java está instalado.

En Ubuntu Server se puede consultar el valor de las variables de entorno mediante el comando "env". Para configurar nuevas variables o para modificar las existentes se debe modificar el fichero "/etc/environment".

Las variables principales a configurar son la variable PATH, la cual debe contener también el directorio de java, y la variable JAVA\_HOME, que debe crearse.

En vez de utilizar el fichero "/etc/environment" para configurar las variables de entorno, se ejecuta un paquete muy útil para hacer toda la configuración necesaria. El paquete es un paquete de apt que tan sólo realiza la configuración de variables de entorno de java, sin llegar a instalar nada.

El paquete es: "oracle-java11-set-default" y ya se incluye en la instalación del paquete general (oracle-java11-installer).

```
sudo apt install oracle-java11-set-default
```

Después de instalar este paquete, hay que reiniciar el sistema, y comprobar el valor de las variables de entorno mencionadas.



Para comprobar si se han configurado las variables se debe ejecutar el comando "env". Entre otras variables, se debe comprobar que las variables JAVA\_HOME y PATH apuntan a los directorios donde se encuentra instalado Java.

```
JAVA_HOME=/usr/lib/jvm/java-11-oracle
J2SDKDIR=/usr/lib/jvm/java-11-oracle
XDG_VTNR=1
XDG_SESSION_ID=1
DERBY_HOME=/usr/lib/jvm/java-11-oracle/db
HUSHLOGIN=FALSE
USER=linkia
PWD=/home/linkia
HOME=/home/linkia
JOURNAL_STREAM=9:21499
J2REDIR=/usr/lib/jvm/java-11-oracle
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/flatpak/desktop
MAIL=/var/mail/linkia
SHELL=/bin/bash
TERM=linux
SHLVL=1
XDG_SEAT=seat0
LOGNAME=linkia
XDG_RUNTIME_DIR=/run/user/1000
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-11-oracle/bin:/usr/lib/jvm/java-11-oracle/db/bin
LESSOPEN=| /usr/bin/lesspipe %s
_=usr/bin/env
```

### 1.1.2. Instalación de paquetes Tomcat

Con Java instalado y las variables de entorno configuradas ya se puede proceder a instalar Tomcat.

Para realizar la descarga de Tomcat, hay que ejecutar el siguiente comando:

```
cd /tmp
wget http://mirrors.sonic.net/apache/tomcat/tomcat-9/v9.0.12/bin/apache-tomcat-9.0.12.tar.gz
```

En el ejemplo se ha utilizado el directorio /tmp como directorio donde se guardará el archivo descargado.

El siguiente paso es descomprimir el paquete descargado.

```
tar -xzf apache-tomcat-9.0.12.tar.gz
```

Una vez descomprimido el fichero, hay que crear un directorio para los archivos Tomcat y mover a este directorio los archivos descomprimidos en el paso anterior.

```
sudo mv apache-tomcat-9.0.12 /opt/tomcat9
```

Seguidamente hay que crear un usuario tomcat9, para que sea el propietario del contenido del directorio /opt/tomcat9.

```
sudo useradd -r tomcat9 --shell /bin/false
```



Con el siguiente comando, se asigna los permisos de propietario sobre el directorio al usuario tomcat9.

```
sudo chown -R tomcat9 /opt/tomcat9
```

### 1.1.3. Configurar el servicio tomcat9

Después de extraer el paquete Tomcat, hay que ejecutar el comando correspondiente para abrir el archivo de configuración para el usuario por defecto.

```
sudo nano /opt/tomcat9/conf/tomcat-users.xml
```

Modificando el contenido de este archivo se creará un usuario con su correspondiente contraseña. Hay que añadir el siguiente contenido antes de la etiqueta `</tomcat-users>`.

```
<role rolename="manager-gui"/>
```

```
<role rolename="admin-gui"/>
```

```
<user username="admin" password="password_here" roles="manager-gui,admin-gui"/>
```

El siguiente paso, es crear una cuenta en el servidor para que Tomcat funcione como servicio. Para ello es necesario crear un archivo llamado `tomcat.service` y añadir el contenido necesario para que el servicio Tomcat funcione correctamente.

Para crear el archivo, el comando es el siguiente:

```
sudo nano /etc/systemd/system/tomcat.service
```

El contenido a añadir en este archivo, es el siguiente:

```
[Unit]
Description=Tomcat9
After=network.target

[Service]
Type=forking
User=tomcat9
Group=tomcat9

Environment=CATALINA_PID=/opt/tomcat9/tomcat9.pid
Environment=CATALINA_HOME=/opt/tomcat9
Environment=CATALINA_BASE=/opt/tomcat9

ExecStart=/opt/tomcat9/bin/startup.sh
ExecStop=/opt/tomcat9/bin/shutdown.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target
```



Después de guardar y salir del fichero, hay que ejecutar los siguientes comandos para cargar el servicio:

```
sudo systemctl daemon-reload
sudo systemctl start tomcat.service
sudo systemctl restart tomcat.service
sudo systemctl enable tomcat.service
```

Para saber el estado en el que está el servicio de Tomcat, se utilizará el siguiente comando:

```
sudo systemctl status tomcat.service
```

Si el servicio se inicia correctamente, la ejecución del comando anterior lo mostrará:

```
linkia@linkia-server:~$ sudo systemctl status tomcat.service
• tomcat.service - Tomcat9
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2018-10-17 07:55:34 UTC; 6h ago
     Main PID: 887 (java)
       Tasks: 43 (limit: 2320)
    CGroup: /system.slice/tomcat.service
           └─887 /usr/bin/java -Djava.util.logging.config.file=/opt/tomcat9/conf/logging.properties

oct 17 07:55:34 linkia-server systemd[1]: Starting Tomcat9...
oct 17 07:55:34 linkia-server startup.sh[761]: Tomcat started.
oct 17 07:55:34 linkia-server systemd[1]: Started Tomcat9.
```

Tal y como está configurado el servidor Tomcat, sólo es accesible desde la propia máquina donde está el servicio, es decir como localhost. Para abrir el acceso desde otras máquinas, hay que hacer una modificación en las reglas del firewall.

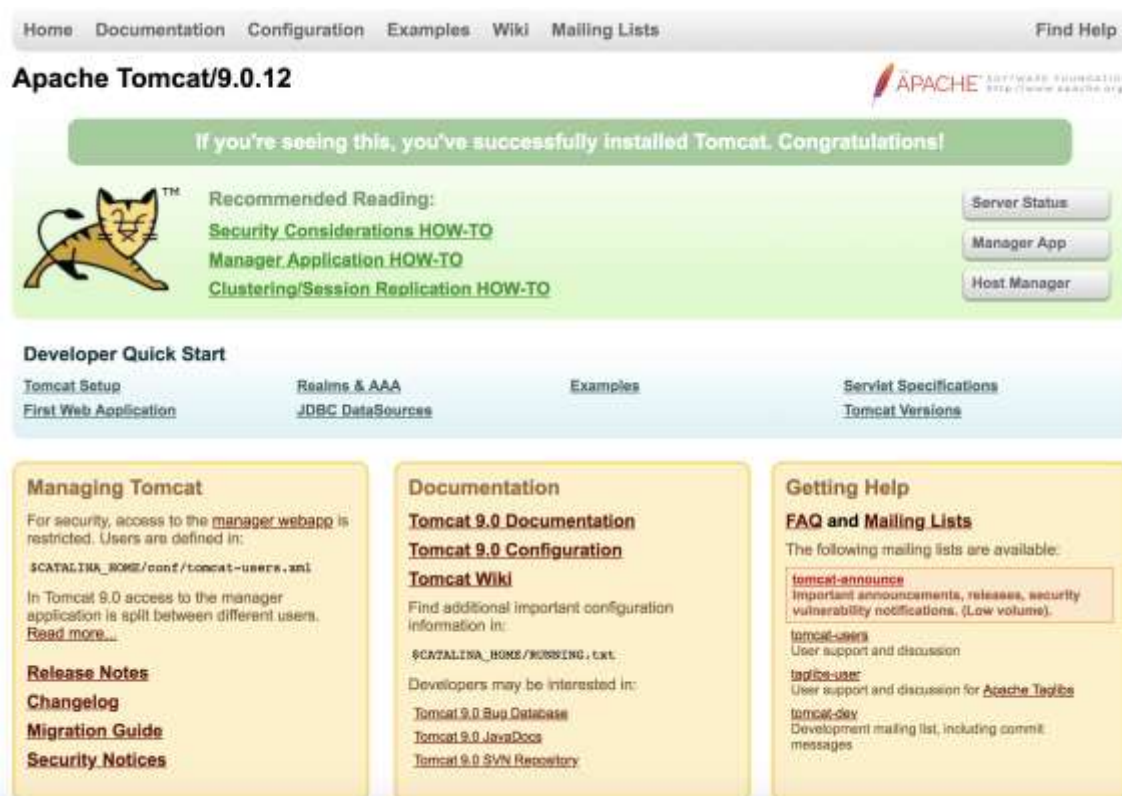
```
sudo ufw allow 8080
```

Ahora ya se puede acceder a la página por defecto del servidor Tomcat desde el navegador, utilizando tanto localhost como la IP del servidor. El puerto que utiliza tomcat es el 8080, por lo tanto la URL a utilizar es:

<http://localhost:8080>

[http://IP\\_SERVIDOR:8080](http://IP_SERVIDOR:8080)





### 1.1.4. Interfaz web de configuración de Tomcat

Para poder gestionar Tomcat de forma práctica, Tomcat proporciona dos interfaces web: Manager y Host Manager.

Por defecto, las nuevas versiones de Tomcat, tienen restringido el acceso a estas aplicaciones de gestión web de Tomcat a conexiones que no se realicen desde la propia máquina que tiene el servicio instalado. Para eliminar esta restricción, hay que modificar el fichero context.xml, tanto del Manager como del Host manager.

Para la aplicación Manager, el archivo que hay que modificar es: /opt/tomcat/webapps/manager/META-INF/context.xml.

```
sudo nano /opt/tomcat9/webapps/manager/META-INF/context.xml
```

Para la aplicación Host Manager, el archivo que hay que modificar es: /opt/tomcat/webapps/host-manager/META-INF/context.xml.

```
sudo nano /opt/tomcat9/webapps/host-manager/META-INF/context.xml
```



Del contenido de ambos ficheros, hay que comentar las líneas que contienen la restricción de dirección IP para permitir las conexiones desde cualquier máquina. Otra opción es que si sólo se quieren permitir conexiones desde una máquina específica, en lugar de comentar las líneas, se indica la IP de la máquina.

```
<Context antiResourceLocking="false" privileged="true" >
<!--
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
<Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|string)$
-->
</Context>
```

Para que los cambios se hagan efectivos, hay que reiniciar el servicio de tomcat.

```
sudo systemctl restart tomcat
```

Ahora ya se puede acceder tanto al Manager como al Host Manager.

[http://IP\\_SERVIDOR:8080/manager/html](http://IP_SERVIDOR:8080/manager/html)

Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Repegar Expirar sesiones: sin trabajar a 30 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Repegar Expirar sesiones: sin trabajar a 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Repegar Expirar sesiones: sin trabajar a 30 minutos
/hello	Ninguno especificado	Hello, World Application	true	0	Arrancar Parar Recargar Repegar Expirar sesiones: sin trabajar a 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Repegar Expirar sesiones: sin trabajar a 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Repegar Expirar sesiones: sin trabajar a 30 minutos

Para acceder al Manager, el servidor pedirá autenticación de usuario. Hay que utilizar el usuario creado en el archivo tomcat-users.xml.

La aplicación Manager, se utiliza para gestionar las aplicaciones Java. Desde aquí se inician, paran, reinician y despliegan las aplicaciones. También se pueden realizar diagnósticos de las aplicaciones. Al final de la página, hay información sobre el servidor.

En aplicación Manager, se pueden observar las aplicaciones desplegadas:

- docs: conjunto de documentación sobre Tomcat accesible vía web.
- examples: ejemplos de aplicaciones con el código visible y ejecutable. Se muestran ejemplos de servlets y JSPs.
- host-manager: gestor de máquina virtual de tomcat.



- manager: la propia interfaz web de configuración de aplicaciones.

También en la parte inferior se puede observar un formulario para poder desplegar aplicaciones nuevas mediante archivos WAR (Web Archive). Los archivos WAR son paquetes que incluyen todos los ficheros de la aplicación web.

The screenshot shows the 'Desplegar' (Deploy) section of the Tomcat Host Manager interface. It has a yellow header with the title 'Desplegar' and a subtitle 'Desplegar directorio o archivo WAR localizado en servidor'. Below this, there are four input fields: 'Trayectoria de Contexto (opcional):', 'Version (for parallel deployment):', 'URL de archivo de Configuración XML:', and 'URL de WAR o Directorio:'. A 'Desplegar' button is located to the right of the last field. Below these fields is another yellow section titled 'Archivo WAR a desplegar'. It contains a text input field with the placeholder 'Seleccione archivo WAR a cargar', a 'Seleccionar archivo' button, and a status indicator 'Ningún archivo seleccionado'. A second 'Desplegar' button is at the bottom right of this section.

[http://IP\\_SERVIDOR:8080/host-manager/html](http://IP_SERVIDOR:8080/host-manager/html)

The screenshot shows the main interface of the Tomcat Host Manager. At the top left is the Tomcat logo (a cat), and at the top right is the Apache Software Foundation logo. The title 'Gestor de Máquina Virtual de Tomcat' is centered. Below the title is a 'Mensaje:' field with an 'OK' button. A navigation bar contains links: 'Gestor de Máquina', 'Lista de Máquinas Virtuales', 'Ayuda de Gestor de Máquina HTML (En breves)', 'Ayuda de Gestor de Máquina', and 'Estado de Servidor'. The main content area has a table with columns 'Nombre de Máquina', 'Alisos de Máquina', and 'Comandos'. The first row shows 'localhost' and 'Instalado: Gestor de Máquinas - comentarios desactivados'. Below the table is a section titled 'Añadir Máquina Virtual' with a 'Máquina' label. It contains several input fields: 'Nombre:', 'Alias:', 'App base:', 'AutoDeploy' (checked), 'DeployOnStartup' (checked), 'DeployXML' (checked), 'UnpackWARs' (checked), 'App de Gestor' (checked), and 'CopyXML' (unchecked). An 'Añadir' button is at the bottom right.

La aplicación Host Manager permite añadir hosts virtuales para utilizar en las aplicaciones.





## 1.2. Despliegue de una aplicación web de ejemplo

Además de examinar las aplicaciones web de ejemplo instaladas con el paquete examples, se pueden desplegar nuevas aplicaciones.

Se puede obtener una aplicación web de internet en formato WAR y desplegarla directamente desde el manager web de Tomcat.

A continuación se muestra el despliegue de una aplicación básica (Hello World) obtenida desde <http://mindprod.com/jgloss/warfile.html>

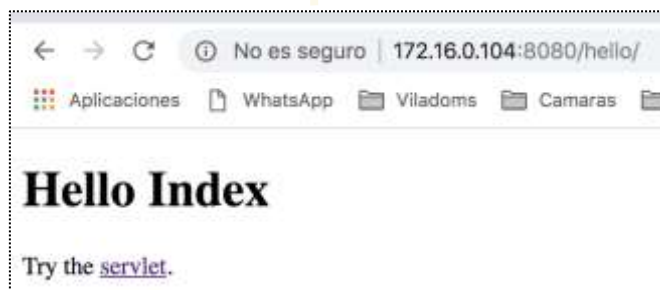
De dicha página se obtiene un fichero WAR. Desde el web manager se despliega el fichero:

Archivo WAR a desplegar	
Seleccione archivo WAR a cargar	<input type="button" value="Seleccionar archivo"/> hello.war
<input type="button" value="Desplegar"/>	

Después de hacer click en Desplegar, la aplicación se pone en marcha tal como se puede ver en la lista de aplicaciones desplegadas:

Aplicaciones					
Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <small>Expirar sesiones: sin trabajar a: 30 minutos</small>
/docs	Ninguno especificado	Tomcat Documentation	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <small>Expirar sesiones: sin trabajar a: 30 minutos</small>
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <small>Expirar sesiones: sin trabajar a: 30 minutos</small>
/hello	Ninguno especificado	Hello, World Application	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <small>Expirar sesiones: sin trabajar a: 30 minutos</small>
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <small>Expirar sesiones: sin trabajar a: 30 minutos</small>
/manager	Ninguno especificado	Tomcat Manager Application	true	1	<input type="button" value="Arrancar"/> <input type="button" value="Parar"/> <input type="button" value="Recargar"/> <input type="button" value="Replegar"/> <small>Expirar sesiones: sin trabajar a: 30 minutos</small>

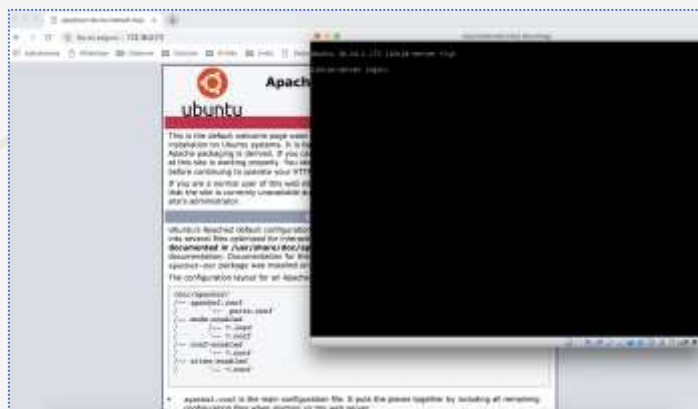
Haciendo clic en la aplicación, o mediante la ruta correspondiente se puede acceder a la aplicación web.



Aunque este ejemplo de aplicación web es muy básico y de hecho no se ejecuta un código con resultados dinámicos, lo importante era mostrar el procedimiento para el despliegue de aplicaciones web en Tomcat mediante ficheros WAR.



### Recursos/Enlaces



→ Video instalación Tomcat

### Conceptos clave

- **Tomcat:** es un contenedor de servlets que se utiliza en la referencia oficial de la implementación para Java Servlet y JavaServer Pages (JSP). Las especificaciones Java Servlet y JavaServer Pages son desarrolladas por Sun Microsystems cuyas especificaciones vienen dadas por la JCP (Java Community Process).
- **Servlet:** es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Este tipo de servlets son la contraparte Java de otras tecnologías de contenido dinámico Web, como PHP y ASP.NET.
- **JSP:** JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML y XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.



## Test de autoevaluación

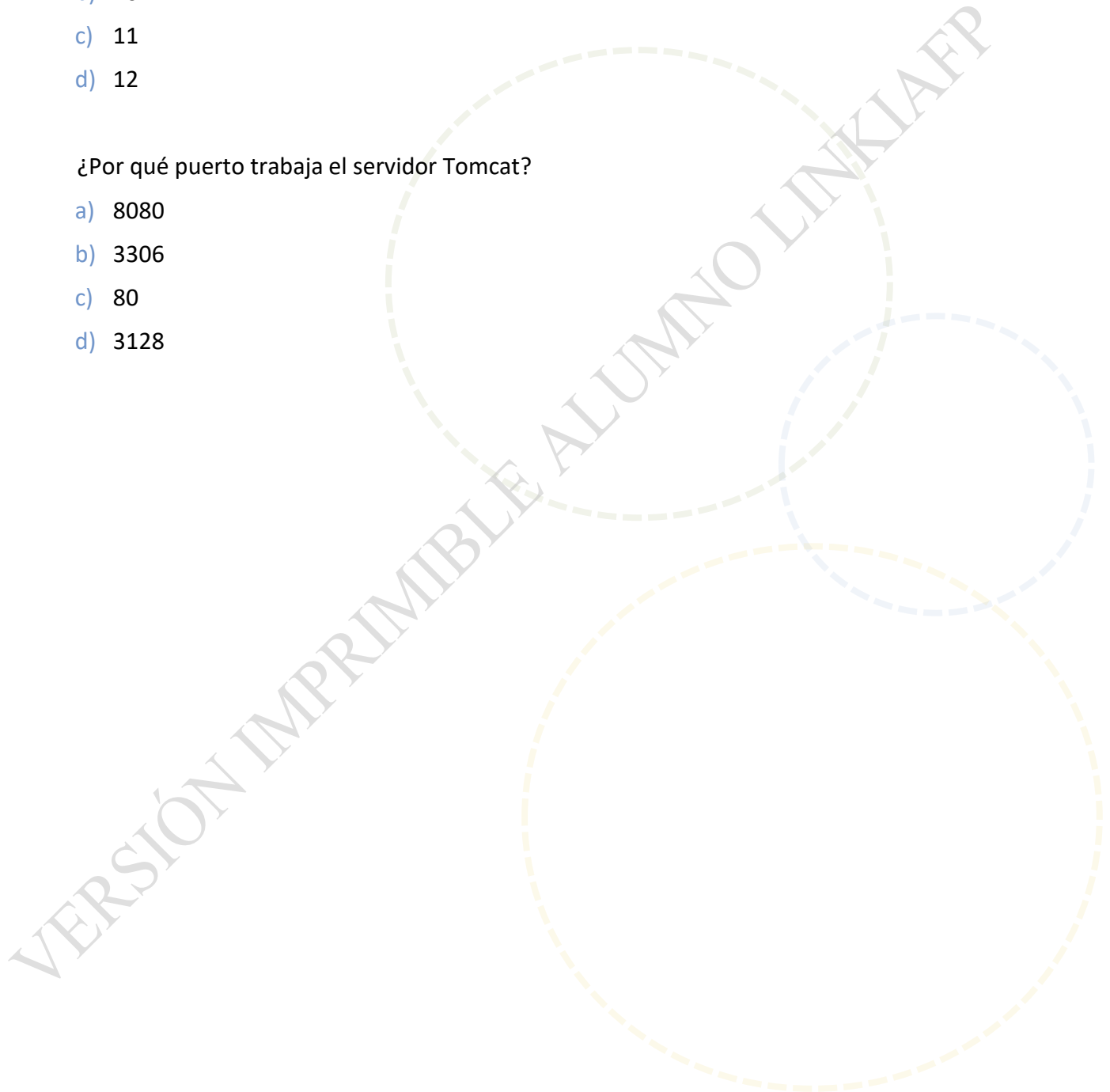
---

¿Qué versión de Java se ha instalado en el ejemplo de estos apuntes?

- a) 9
- b) 10
- c) 11
- d) 12

¿Por qué puerto trabaja el servidor Tomcat?

- a) 8080
- b) 3306
- c) 80
- d) 3128





## Ponlo en práctica

### Actividad 1

1. En una máquina virtual con Ubuntu Server instala un servidor Tomcat
2. Configura el servidor para poder utilizar el Manager y el Host Manager
3. Despliega una aplicación de ejemplo de la siguiente página:

<http://mindprod.com/jgloss/warfile.html>



SOLUCIÓN

### Actividad 2

Completa la siguiente frase con las palabras del listado:

Un \_\_\_ es un programa \_\_\_ que extiende las capacidades de un \_\_\_. Habitualmente implementa aplicaciones albergadas en servidores \_\_\_. Un servlet es código Java que produce resultados en formato \_\_\_, para que después el servidor web pueda enviar al \_\_\_ mediante http.

cliente	servidor	servlet	html	java	web
---------	----------	---------	------	------	-----



SOLUCIÓN



VERSIÓN IMPRIMIBLE ALUMNO LINKIAFP

VERSIÓN IMPRIMIBLE ALUMNO LINKIAFP



**Linkia FP**

Formación Profesional Oficial a Distancia

